In [1]:

```
 %config InlineBackend.figure_format = 'retina'
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn import svm
import seaborn as sns
import random
import scipy.io as sio
from sklearn import neighbors
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn import tree
import pandas as pd
import csv
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

svm

In [2]:

```
C_list = [1, 10, 100, 1000, 10000]
gamma_list = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]

def SVM(X_train, Y_train, X_test, Y_test):
    # Create a SVM classifier with RBF kernel.
    estimator = svm.SVC(kernel='rbf')
    # Create a grid searcher with 5-fold cross-validation.
    param_grid = {'C': C_list, 'gamma': gamma_list}
    grid_search = GridSearchCV(estimator, param_grid, cv=5, return_
train_score=True)
    # Use the classifier to fit the training set.
    grid_search.fit(X_train, Y_train)
    # Get the best hyper-parameters
    cross_val_errors = 1 - grid_search.cv_results_['mean_test_score
'].reshape(-1,1)
    training_errors = 1 - grid_search.cv_results_['mean_train_score
']
    best_C = grid_search.best_params_['C']
    best_gamma = grid_search.best_params_['gamma']
    training_error = training_errors[grid_search.best_index_]
    min_cv_err = 1 - grid_search.best_score_
    # Test
    Y_pred = grid_search.best_estimator_.predict(X_test)
    test_error = 1 - accuracy_score(Y_test, Y_pred)
    return([cross_val_errors, best_C, best_gamma, training_error, m
in_cv_err, test_error])
```

svm heatmap

In [3]:

```
# SVM: Draw the heatmap of training errors.
def draw_heatmap_SVM(training_errors, gamma_list, C_list):
    #plt.figure(figsize = (5,10))
    ax = sns.heatmap(training_errors, annot=True, fmt='.3f',xticklabels=gamma_list, yticklabels=C_list)
    ax.collections[0].colorbar.set_label("error")
    ax.set(xlabel = '$\gamma$', ylabel='$C$')
    bottom, top = ax.get_ylim()
    ax.set_ylim(bottom + 0.5, top - 0.5)
    plt.title('Training error w.r.t $C$ and $\gamma$')
    plt.show()
```

knn

In [4]:

```
#KNN
k_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
def KNN(X_train, Y_train, X_test, Y_test):
    # Create a k-NN classifier.
    estimator = neighbors.KNeighborsClassifier()
    # Create a grid searcher with 5-fold cross-validation.
    param_grid = {'n_neighbors': k_list}
    grid_search = GridSearchCV(estimator, param_grid, cv=5, return_train_score=True)
    # Use the grid searcher to fit the training set.
    grid_search.fit(X_train, Y_train)
    # Get the best hyper-parameters
    cross_val_errors = 1 - grid_search.cv_results_['mean_test_score'].reshape(-1,1)
    training_errors = 1 - grid_search.cv_results_['mean_train_score']
    best_k = grid_search.best_params_['n_neighbors']
    training_error = training_errors[grid_search.best_index_]
    min_cv_err = 1 - grid_search.best_score_
    # Test
    Y_pred = grid_search.best_estimator_.predict(X_test)
    test_error = 1 - accuracy_score(Y_test, Y_pred)
    return([cross_val_errors, best_k, training_error, min_cv_err, test_error])
```

knn heatmap

In [5]:

```
def draw_heatmap_KNN(errors, D_list, title):
    #plt.figure(figsize = (5,10))
    ax = sns.heatmap(errors, annot=True, fmt='.3f', yticklabels=D_list, xticklabels=[])
    ax.collections[0].colorbar.set_label('error')
    ax.set(ylabel='k')
    bottom, top = ax.get_ylim()
    ax.set_ylim(bottom + 0.5, top - 0.5)
    plt.title(title)
    plt.show()
```

decision tree

In [6]:

```
#decision tree
D_list = [1, 2, 3, 4, 5]
def DecisionTree(X_train, Y_train, X_test, Y_test):
    # Create a decision tree classifier.
    estimator = tree.DecisionTreeClassifier("entropy", random_state
= 1)
    # Create a grid searcher with cross-validation.
    param_grid = {'max_depth': D_list}
    grid_search = GridSearchCV(estimator, param_grid, cv=5, return_
train_score=True)
    # Use the grid searcher to fit the training set.
    grid_search.fit(X_train, Y_train)
    # Get the best hyper-parameters
    cross_val_errors = 1 - np.reshape(grid_search.cv_results_['mean
_test_score'], (5,1))
    training_errors = 1 - grid_search.cv_results_['mean_train_score
']
    best_max_depth = grid_search.best_params_['max_depth']
    training_error = training_errors[grid_search.best_index_]
    min_cv_err = 1 - grid_search.best_score_
    # Test
    Y_pred = grid_search.best_estimator_.predict(X_test)
    test_error = 1 - accuracy_score(Y_test, Y_pred)
    return([cross_val_errors, best_max_depth, training_error, min_c
v_err, test_error])
```

decision tree heatmap

In [7]:

```
# DT: Draw heatmaps for result of grid search.
def draw_heatmap_DT(errors, D_list, title):
    #plt.figure(figsize = (5,10))
    ax = sns.heatmap(errors, annot=True, fmt='.3f', yticklabels=D_l
ist, xticklabels=[])
    ax.collections[0].colorbar.set_label('error')
    ax.set(ylabel='max depth D')
    bottom, top = ax.get_ylim()
    ax.set_ylim(bottom + 0.5, top - 0.5)
    plt.title(title)
    plt.show()
```

In [8]:
```
def calc_error(X, Y, classifier):
    Y_pred = classifier.predict(X)
    e = 1 - accuracy_score(Y, Y_pred)
    return e
```

first dataset

In [25]:
```
#read in the dataset
pulsar_stars = pd.read_csv('pulsar_stars.csv')
pulsar_stars = pulsar_stars[:-15000]
print(pulsar_stars.isnull().sum())
#pulsar_stars = pulsar_stars
X_and_Y = np.array(pulsar_stars).astype(np.float)
#X_and_Y = np.array(data).astype(np.float) # Load data from file.
np.random.seed(0)                # Set the random seed.
np.random.shuffle(X_and_Y)       # Shuffle the data.
X  = X_and_Y[:, 0:-1]# First column to second last column: Features.
Y  = X_and_Y[:, -1]              # Last column: Labels.
Y[Y==0] = -1                     # Convert labels from {0, 1} to {-1
, 1}.
size = len(X)
print(X.shape)
print(Y.shape)
print(X_and_Y[0])
```

```
 Mean of the integrated profile                  0
 Standard deviation of the integrated profile    0
 Excess kurtosis of the integrated profile       0
 Skewness of the integrated profile              0
 Mean of the DM-SNR curve                        0
 Standard deviation of the DM-SNR curve          0
 Excess kurtosis of the DM-SNR curve             0
 Skewness of the DM-SNR curve                    0
target_class                                     0
dtype: int64
(2898, 8)
(2898,)
[120.9609375   58.79724196   0.23221737  -0.75538739   6.27759197
  31.29680227   5.83319713  35.0768173   -1.        ]
```

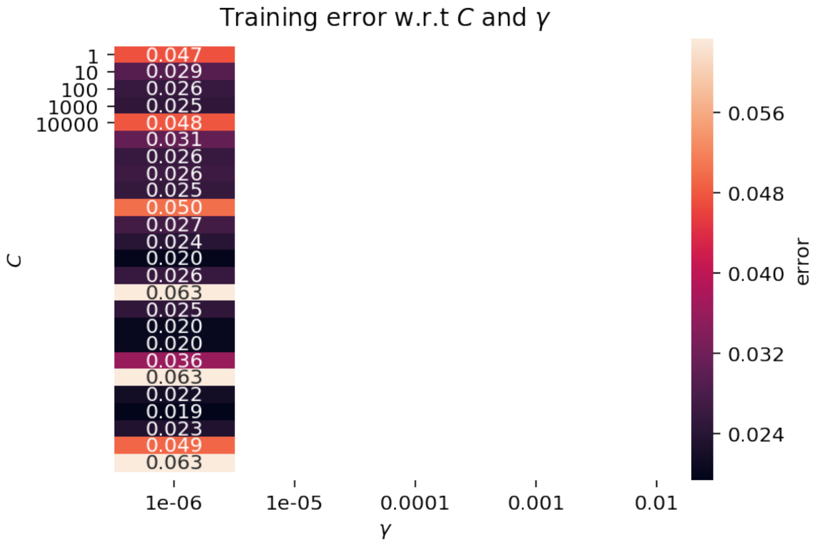
80 train 20 test

In [10]:
```
#80/20
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []
```

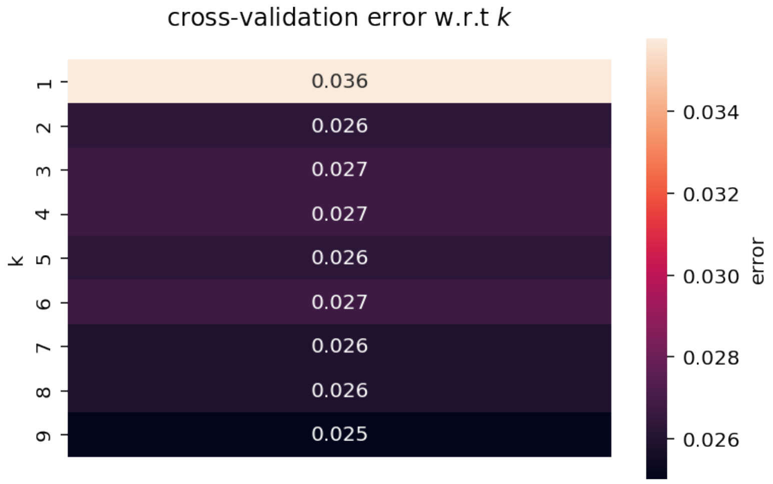
```
sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    #np.random.seed(0)                # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]             # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]              # Last column: Labels.
    Y[Y==0] = -1                    # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.2)
    print(X_train[0])
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validation error w.r.t $k$')


        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)

print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, avg_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn, avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_test_dt,avg_vali_dt)
```

```
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

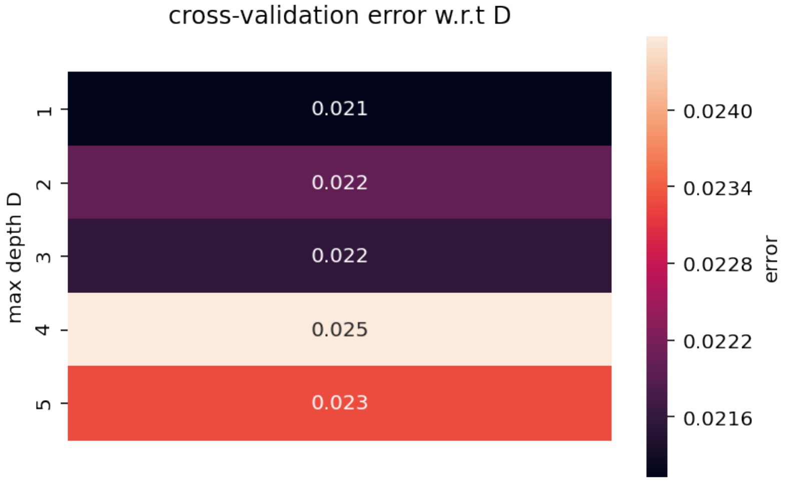
```
---------- trial: 1 ------------
seed: 42096

[ 1.24804688e+02  4.29190842e+01 -7.34590250e-02  3.77548575e-01
  3.14297659e+00  1.82785615e+01  7.47364897e+00  6.60631549e+01]
The shape of X_train is (2318, 8)
The shape of Y_train is (2318,)
The shape of X_test is (580, 8)
The shape of Y_test is (580,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
Training error =  0.016069458619376142
Training accuracy =  0.9839305413806239
sum of training accuracy =  [0.9839305413806239]
Validation error =  0.019403999404185468
Validation accuracy =  0.9805960005958145
sum of Validation accuracy =  [0.9805960005958145]
Test error =  0.03275862068965518
Test accuracy =  0.9672413793103448
sum of Test accuracy =  [0.9805960005958145]
```



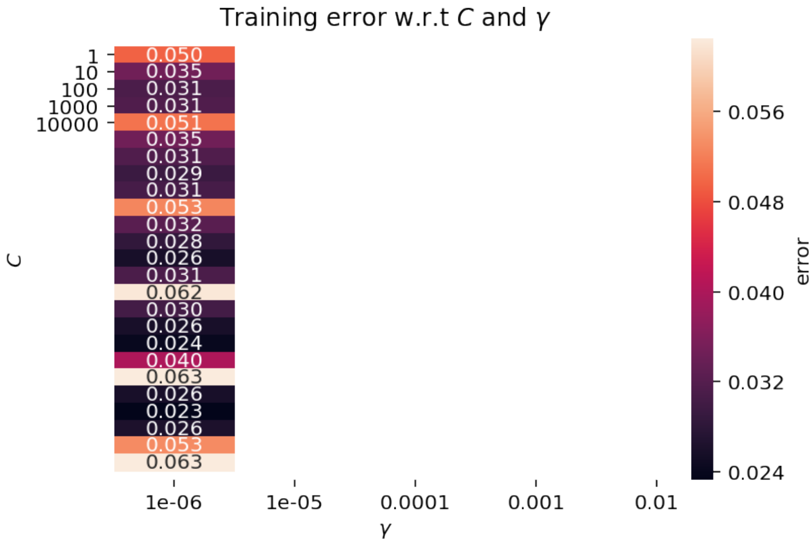
```
classifier: KNN
BEST_K IS 9
Training error =  0.024373962322304488
Training accuracy =  0.9756260376776955
Training accuracy =  0.9756260376776955
sum of training accuracy =  [0.9756260376776955]
Validation error =  0.025012102480077746
Validation accuracy =  0.9749878975199225
sum of Validation accuracy =  [0.9749878975199225]
Test error =  0.03103448275862064
Test accuracy =  0.9689655172413794
sum of Test accuracy =  [0.9689655172413794]
```



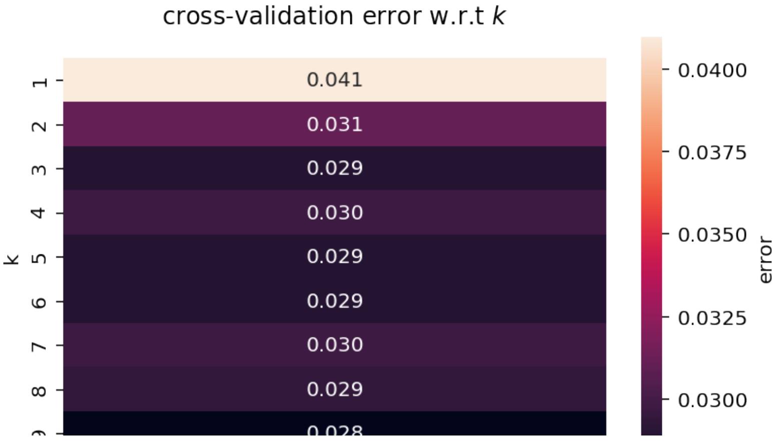
```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.019304890424142962
Training accuracy =  0.980695109575857
sum of training accuracy =  [0.980695109575857]
Validation error =  0.021129999255231957
Validation accuracy =  0.978870000744768
sum of Validation accuracy =  [0.978870000744768]
Test error =  0.02931034482758621
Test accuracy =  0.9706896551724138
sum of Test accuracy =  [0.9706896551724138]
```



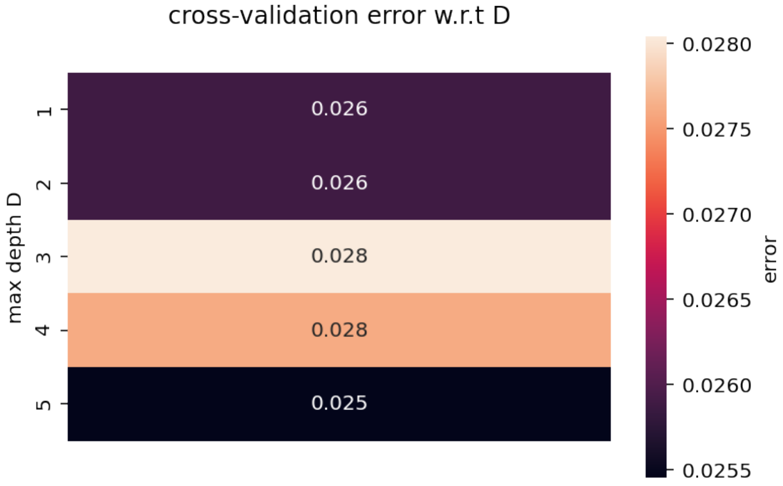
```
---------- trial: 2 ------------
seed: 12809

[102.0703125    42.89470822    0.28022515    0.63586995    2.9590301
  18.21203139    8.59881964   88.98838308]
The shape of X_train is (2318, 8)
The shape of Y_train is (2318,)
The shape of X_test is (580, 8)
The shape of Y_test is (580,)
The shape of Y_test is (580,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
Training error =  0.01887414695987688
Training accuracy =  0.9811258530401231
sum of training accuracy =  [0.9839305413806239, 0.981125853040123
1]
Validation error =  0.023295412229090595
Validation accuracy =  0.9767045877709094
sum of Validation accuracy =  [0.9805960005958145, 0.9767045877709
094]
Test error =  0.015517241379310332
Test accuracy =  0.9844827586206897
sum of Test accuracy =  [0.9805960005958145, 0.9767045877709094]
```



```
classifier: KNN
BEST_K IS 9
Training error =  0.027178708816371414
Training accuracy =  0.9728212911836286
sum of training accuracy =  [0.9756260376776955, 0.972821291183628
6]
Validation error =  0.027612273776718488
Validation accuracy =  0.9723877262232815
sum of Validation accuracy =  [0.9749878975199225, 0.9723877262232
815]
Test error =  0.018965517241379293
Test accuracy =  0.9810344827586207
sum of Test accuracy =  [0.9689655172413794, 0.9810344827586207]
```

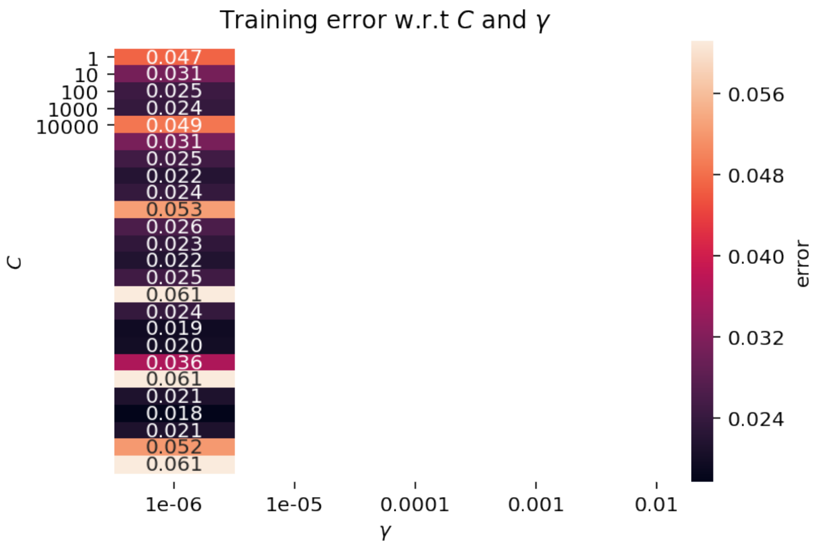


```
classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.015207157540918281
Training accuracy =  0.9847928424590817
sum of training accuracy =  [0.980695109575857, 0.9847928424590817
]
Validation error =  0.025454308482907755
Validation accuracy =  0.9745456915170922
sum of Validation accuracy =  [0.978870000744768, 0.97454569151709
22]
Test error =  0.020689655172413834
Test accuracy =  0.9793103448275862
sum of Test accuracy =  [0.9706896551724138, 0.9793103448275862]
```



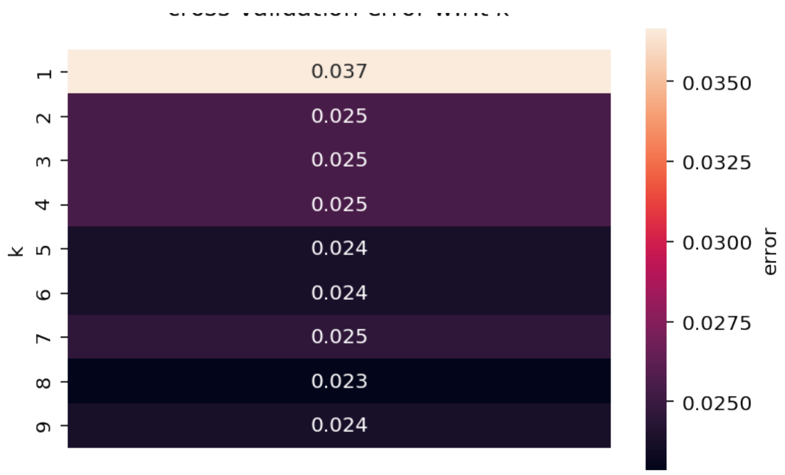
```
---------- trial: 3 ------------
seed: 49100

[97.984375   47.46034272  0.29646352  0.26346842 27.48327759 60.96
164062
  2.00860976  2.61230269]
The shape of X_train is (2318, 8)
The shape of Y_train is (2318,)
The shape of X_test is (580, 8)
The shape of Y_test is (580,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
```

```
Training error =  0.014991233349907O4
Training accuracy =  0.985008766650093
sum of training accuracy =  [0.9839305413806239, 0.981125853040123
1, 0.985008766650093]
Validation error =  0.017686378193192853
Validation accuracy =  0.9823136218068071
sum of Validation accuracy =  [0.9805960005958145, 0.9767045877709
094, 0.9823136218068071]
Test error =  0.02758620689655178
Test accuracy =  0.9724137931034482
sum of Test accuracy =  [0.9805960005958145, 0.9767045877709094, 0
.9823136218068071]
```

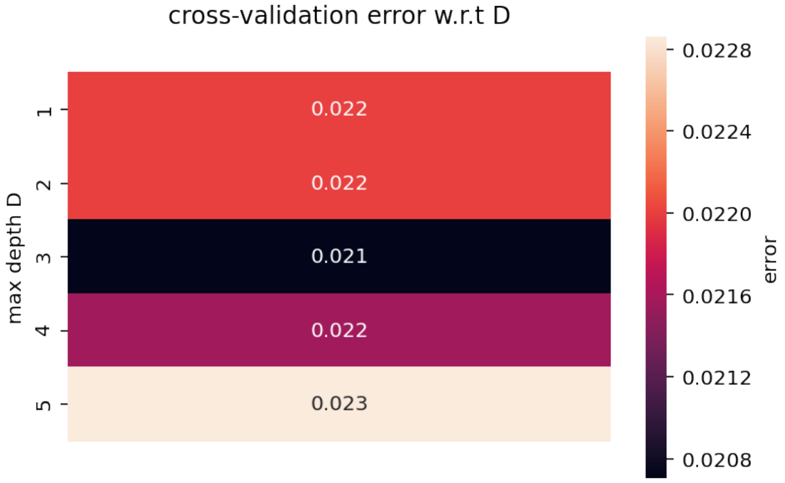


```
classifier: KNN
BEST_K IS 8
Training error =  0.022756711648450056
Training accuracy =  0.9772432883515499
sum of training accuracy =  [0.9756260376776955, 0.972821291183628
6, 0.9772432883515499]
Validation error =  0.022865308706337 96
Validation accuracy =  0.977134691293662
sum of Validation accuracy =  [0.9749878975199225, 0.9723877262232
815, 0.977134691293662]
Test error =  0.03275862068965518
Test accuracy =  0.9672413793103448
sum of Test accuracy =  [0.9689655172413794, 0.9810344827586207, 0
.9672413793103448]
```

cross-validation error w.r.t k



```
classifier: DecisionTree
BEST_max_depth D is 3
Training error =  0.0176877560574209
Training accuracy =  0.9823122439425791
sum of training accuracy =  [0.980695109575857, 0.9847928424590817
, 0.9823122439425791]
Validation error =  0.020706412452521028
Validation accuracy =  0.979293587547479
sum of Validation accuracy =  [0.978870000744768, 0.97454569151709
22, 0.979293587547479]
Test error =  0.02931034482758621
Test accuracy =  0.9706896551724138
sum of Test accuracy =  [0.9706896551724138, 0.9793103448275862, 0
.9706896551724138]
```



```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.9833550536902799 0.979
871403391177 0.979871403391177
sum_svm= 2.943097860472634
avg_train_knn, avg_test_knn, avg_vali_knn = 0.9752302057376246 0.9
724137931034483 0.9748367716789553
sum_knn = 2.9224807705200284
sum_knn = 2.92214007705200204
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9826000653258392 0.9735
632183908045 0.9775697599364465
sum_dt = 2.93373304365309
```

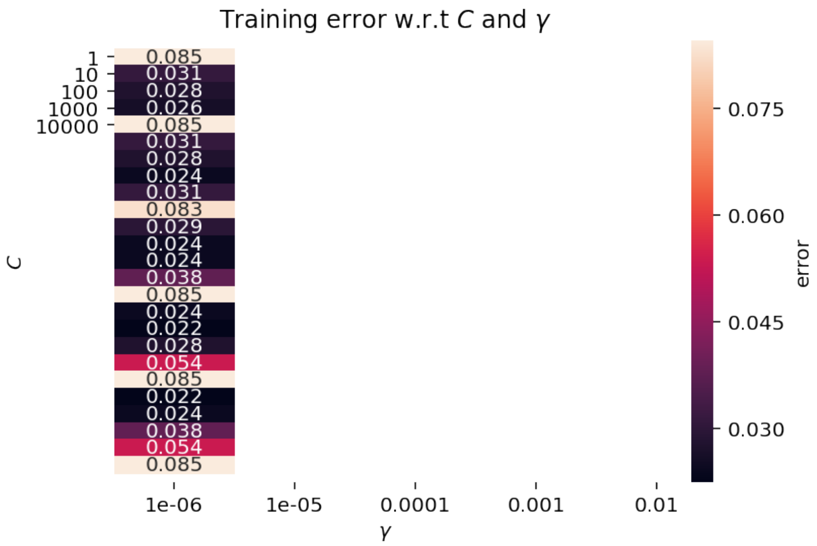
20 train 80 test

In [13]:

```
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                    # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]                  # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]                   # Last column: Labels.
    Y[Y==0] = -1                         # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.8)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[5])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validatio
n error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)
```

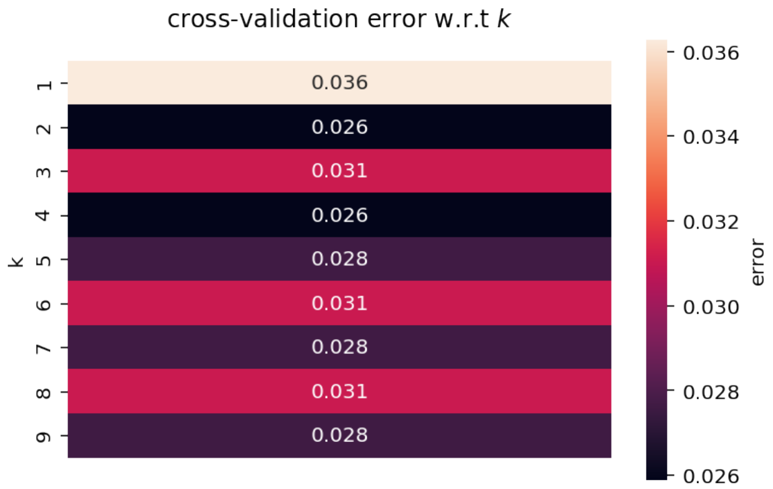
```
avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

```
---------- trial: 1 ------------
seed: 43644

The shape of X_train is (579, 8)
The shape of Y_train is (579,)
The shape of X_test is (2319, 8)
The shape of Y_test is (2319,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-05
Training error =  0.022452893423698606
Training accuracy =  0.9775471065763014
sum of training accuracy =  [0.9775471065763014]
Validation error =  0.022443778110944557
Validation accuracy =  0.9775562218890554
sum of Validation accuracy =  [0.9775562218890554]
Test error =  0.03277274687365239
Test accuracy =  0.9672272531263476
sum of Test accuracy =  [0.9672272531263476]
```

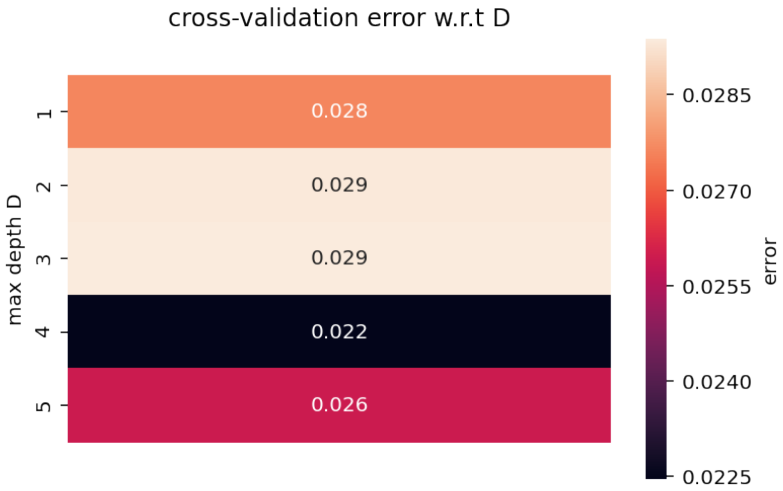


```
classifier: KNN
BEST_K IS 4
Training error =  0.023315893349221795
Training accuracy =  0.9766841066507782
sum of training accuracy =  [0.9766841066507782]
Validation error =  0.025892053973013418
Validation accuracy =  0.9741079460269866
```

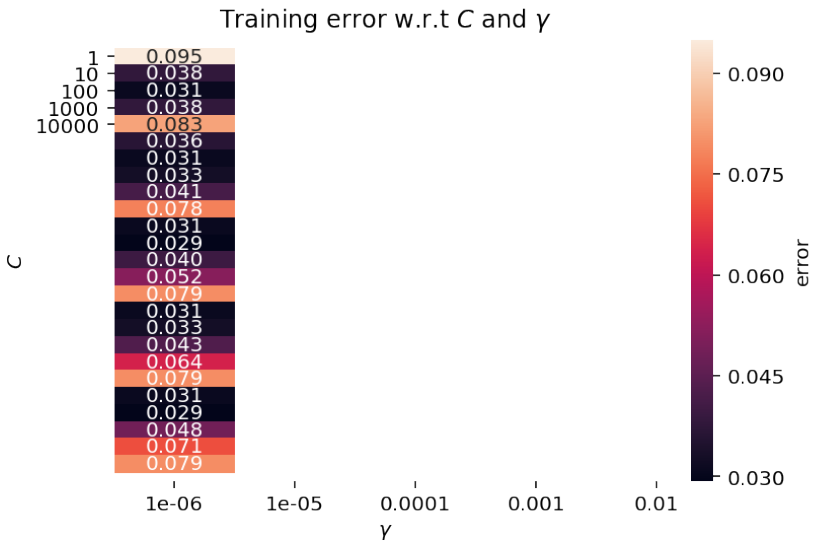
```
sum of Validation accuracy =  [0.9741079460269866]
Test error =  0.031910306166451075
Test accuracy =  0.9680896938335489
sum of Test accuracy =  [0.9680896938335489]
```



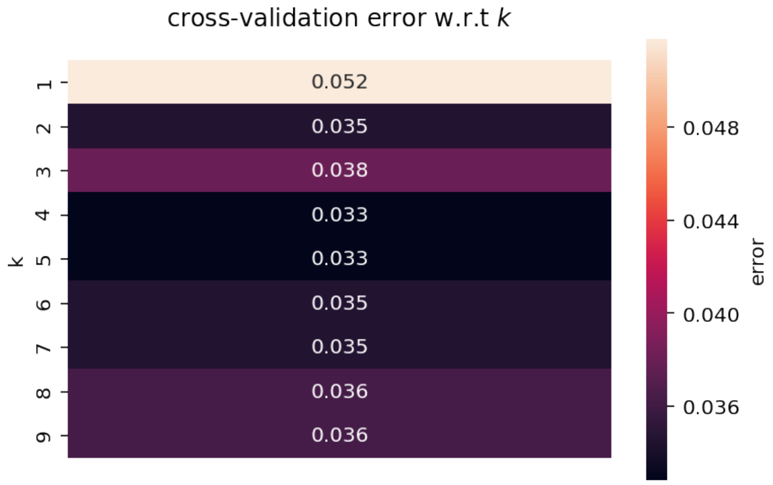
```
classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.010362515826320084
Training accuracy =  0.9896374841736799
sum of training accuracy =  [0.9896374841736799]
Validation error =  0.022458770614692702
Validation accuracy =  0.9775412293853073
sum of Validation accuracy =  [0.9775412293853073]
Test error =  0.025442000862440755
Test accuracy =  0.9745579991375592
sum of Test accuracy =  [0.9745579991375592]
```



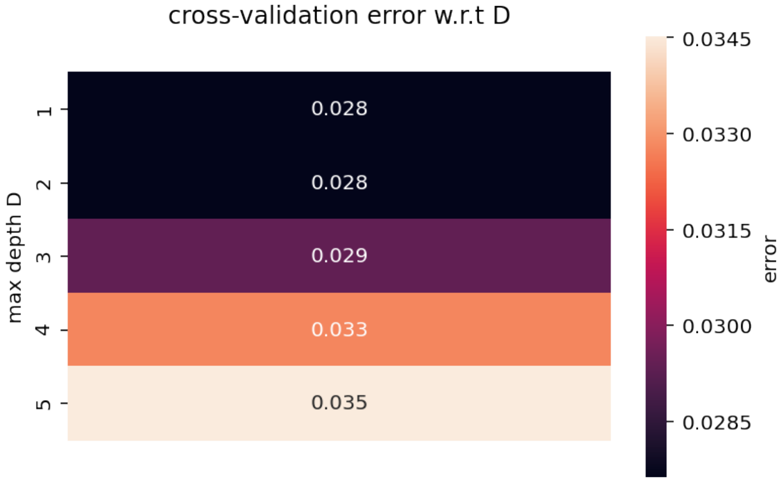
```
---------- trial: 2 ------------
seed: 15561

The shape of X_train is (579, 8)
The shape of Y_train is (579,)
The shape of X_test is (2319, 8)
The shape of Y_test is (2319,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
Training error =  0.015543308259477251
Training accuracy =  0.9844566917405227
sum of training accuracy =  [0.9775471065763014, 0.984456691740522
7]
Validation error =  0.029355322338830647
Validation accuracy =  0.9706446776611694
sum of Validation accuracy =  [0.9775562218890554, 0.9706446776611
694]
Test error =  0.02156101768003449
Test accuracy =  0.9784389823199655
sum of Test accuracy =  [0.9672272531263476, 0.9784389823199655]
```



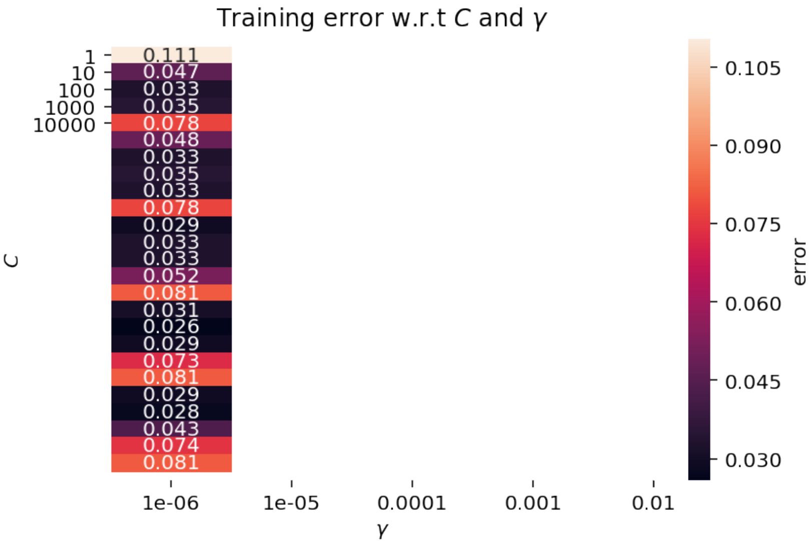
```
classifier: KNN
BEST_K IS 4
Training error =  0.027634616816861524
Training accuracy =  0.9723653831831385
sum of training accuracy =  [0.9766841066507782, 0.972365383183138
5]
Validation error =  0.0328335832083958
Validation accuracy =  0.9671664167916042
sum of Validation accuracy =  [0.9741079460269866, 0.9671664167916
042]
Test error =  0.030616645105648987
Test accuracy =  0.969383354894351
sum of Test accuracy =  [0.9680896938335489, 0.969383354894351]
```



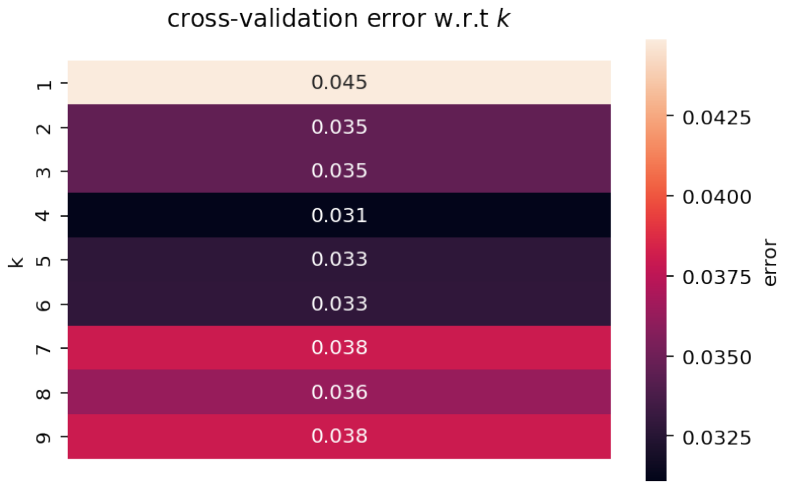
```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.0194300662843524
Training accuracy =  0.9805699337156476
sum of training accuracy =  [0.9896374841736799, 0.980569933715647
6]
Validation error =  0.027631184407796106
Validation accuracy =  0.9723688155922039
sum of Validation accuracy =  [0.9775412293853073, 0.9723688155922
039]
Test error =  0.026735661923242726
Test accuracy =  0.9732643380767573
sum of Test accuracy =  [0.9745579991375592, 0.9732643380767573]
```



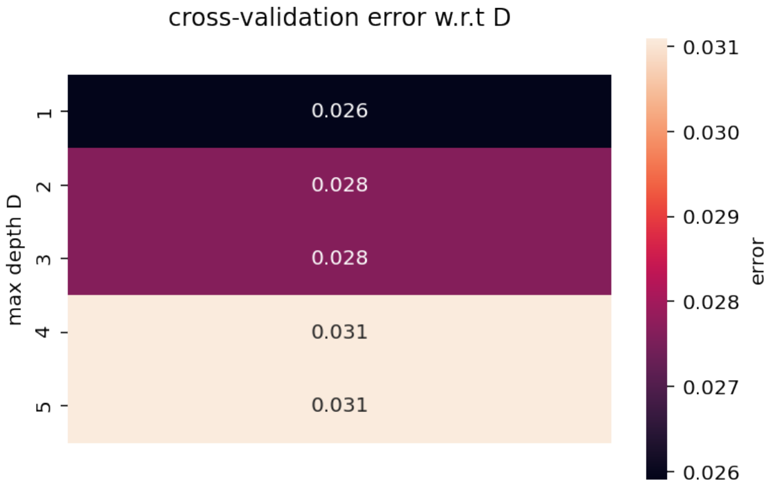
```
---------- trial: 3 ------------
seed: 40518

The shape of X_train is (579, 8)
The shape of Y_train is (579,)
The shape of X_test is (2319, 8)
The shape of Y_test is (2319,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-05
Training error =  0.026337789528561806
Training accuracy =  0.9736622104714382
sum of training accuracy =  [0.9775471065763014, 0.984456691740522
7, 0.9736622104714382]
Validation error =  0.025907046476761564
Validation accuracy =  0.9740929535232384
sum of Validation accuracy =  [0.9775562218890554, 0.9706446776611
694, 0.9740929535232384]
Test error =  0.021992238033635148
Test accuracy =  0.9780077619663649
sum of Test accuracy =  [0.9672272531263476, 0.9784389823199655, 0
.9780077619663649]
```



```
classifier: KNN
BEST_K IS 4
Training error =  0.028929582185149405
Training accuracy =  0.9710704178148506
sum of training accuracy =  [0.9766841066507782, 0.972365383183138
5, 0.9710704178148506]
Validation error =  0.031094452773613335
Validation accuracy =  0.9689055472263867
sum of Validation accuracy =  [0.9741079460269866, 0.9671664167916
042, 0.9689055472263867]
Test error =  0.028460543337645583
Test accuracy =  0.9715394566623544
sum of Test accuracy =  [0.9680896938335489, 0.969383354894351, 0.
9715394566623544]
```



```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.024179824234750846
Training accuracy =  0.9758201757652492
sum of training accuracy =  [0.9896374841736799, 0.980569933715647
6, 0.9758201757652492]
Validation error =  0.025907046476761564
Validation accuracy =  0.9740929535232384
sum of Validation accuracy =  [0.9775412293853073, 0.9723688155922
039, 0.9740929535232384]
Test error =  0.021129797326433833
Test accuracy =  0.9788702026735662
sum of Test accuracy =  [0.9745579991375592, 0.9732643380767573, 0
.9788702026735662]
```



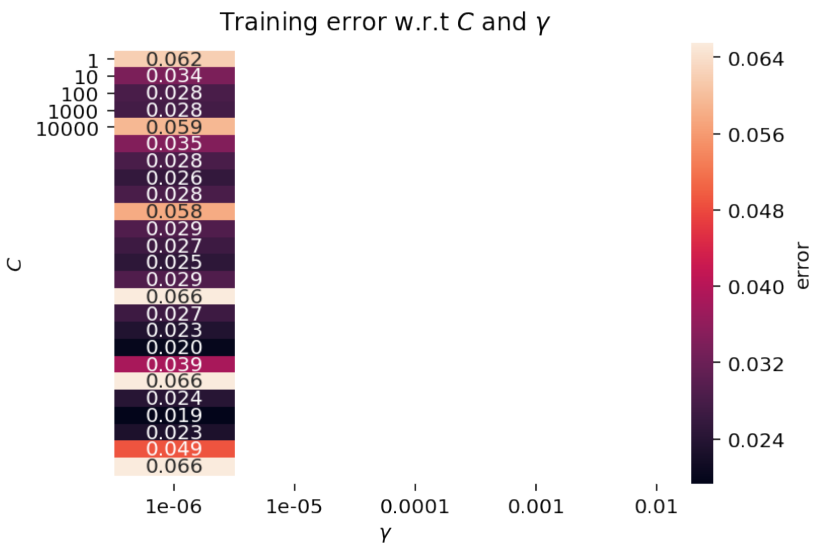
```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.9785553362627541 0.974
5579991375594 0.9740979510244877
sum_svm= 2.927211286424801
avg_train_knn, avg_test_knn, avg_vali_knn = 0.9733733025495891 0.9
696708351300848 0.9700599700149924
sum_knn = 2.913104107694666
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9820091978848589 0.9755
641799626275 0.9746676661669165
sum_dt = 2.9322410440144027
```

50 train 50 test

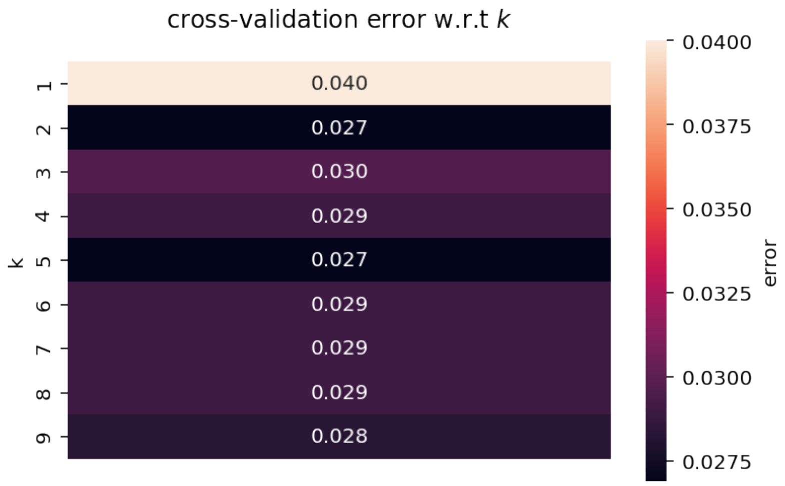
In [14]:

```python
#50/50
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]              # First column to second last column: Features.
    print()
    Y = X_and_Y[:, -1]               # Last column: Labels.
    Y[Y==0] = -1                     # Convert labels from {0, 1} to {-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.5)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[5])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)
        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validation error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, avg_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn, avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

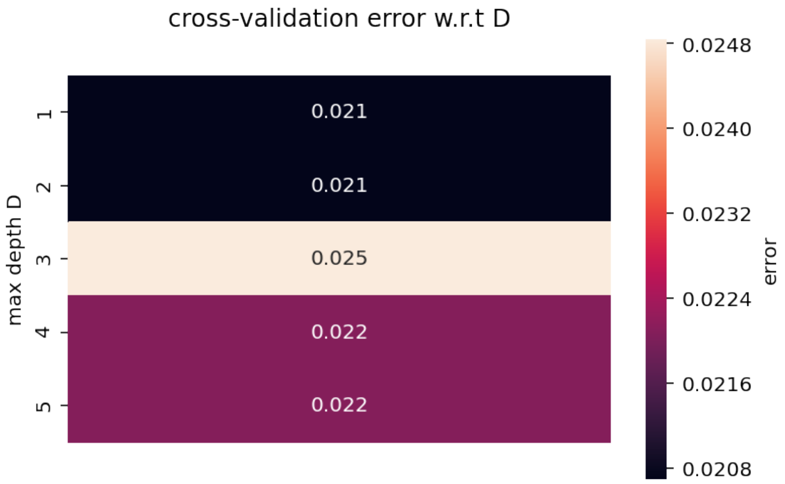
```
---------- trial: 1 ------------
seed: 15883

The shape of X_train is (1449, 8)
The shape of Y_train is (1449,)
The shape of X_test is (1449, 8)
The shape of Y_test is (1449,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
Training error =  0.01777052155544312
Training accuracy =  0.9822294784445569
sum of training accuracy =  [0.9822294784445569]
Validation error =  0.019319890227896486
Validation accuracy =  0.9806801097721035
sum of Validation accuracy =  [0.9806801097721035]
Test error =  0.021394064872325758
Test accuracy =  0.9786059351276742
sum of Test accuracy =  [0.9786059351276742]
```



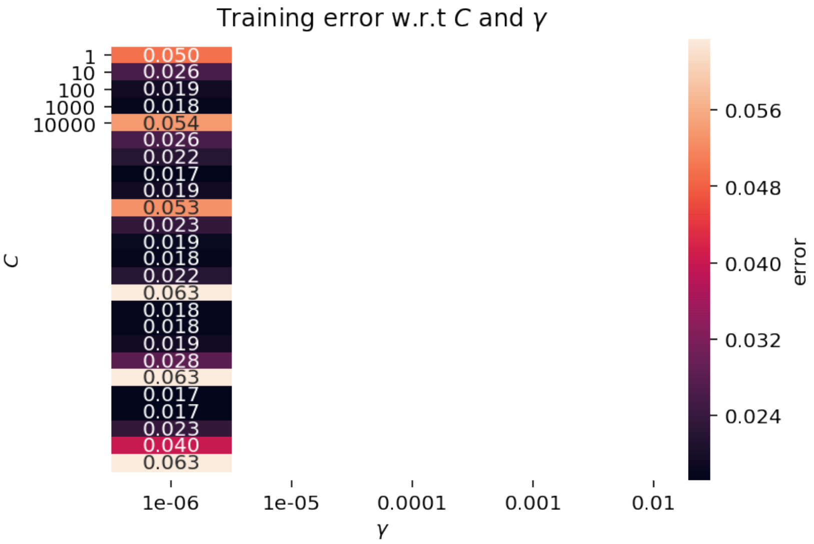
```
classifier: KNN
BEST_K IS 5
Training error =  0.02432670851804475
Training accuracy =  0.9756732914819553
sum of training accuracy =  [0.9756732914819553]
Validation error =  0.02690848347452568
Validation accuracy =  0.9730915165254743
sum of Validation accuracy =  [0.9730915165254743]
Test error =  0.027605244996549372
Test accuracy =  0.9723947550034506
sum of Test accuracy =  [0.9723947550034506]
```



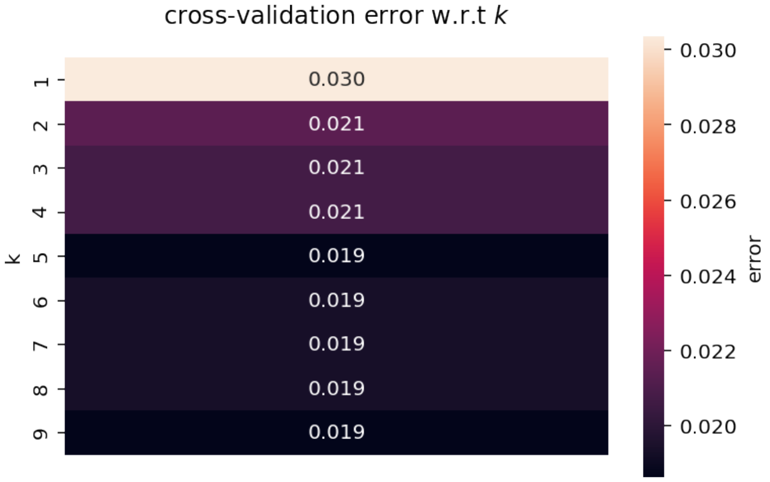
```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.02001353723483379
Training accuracy =  0.9799864627651662
sum of training accuracy =  [0.9799864627651662]
Validation error =  0.020699200572724008
Validation accuracy =  0.979300799427276
sum of Validation accuracy =  [0.979300799427276]
Test error =  0.022774327122153215
Test accuracy =  0.9772256728778468
sum of Test accuracy =  [0.9772256728778468]
```



```
---------- trial: 2 ------------
seed: 18052

The shape of X_train is (1449, 8)
The shape of Y_train is (1449,)
The shape of X_test is (1449, 8)
The shape of Y_test is (1449,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-06
Training error =  0.01673574127517763
Training accuracy =  0.9832642587248224
sum of training accuracy =  [0.9822294784445569, 0.983264258724822
4]
Validation error =  0.01725331106073258
Validation accuracy =  0.9827466889392674
sum of Validation accuracy =  [0.9806801097721035, 0.9827466889392
674]
Test error =  0.031746031746031744
Test accuracy =  0.9682539682539683
sum of Test accuracy =  [0.9786059351276742, 0.9682539682539683]
```

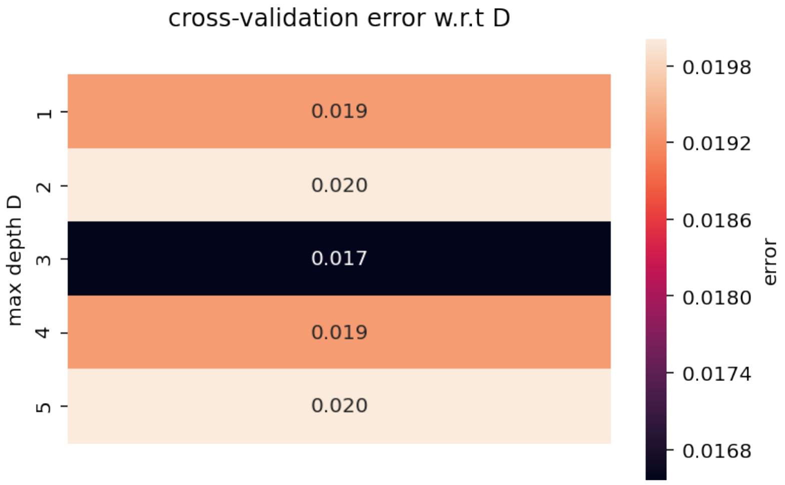


```
classifier: KNN
BEST_K IS 5
Training error =  0.018806194400642573
Training accuracy =  0.9811938055993574
sum of training accuracy =  [0.9756732914819553, 0.981193805599357
4]
Validation error =  0.018632621405560323
Validation accuracy =  0.9813673785944397
sum of Validation accuracy =  [0.9730915165254743, 0.9813673785944
397]
Test error =  0.0331262939958592
Test accuracy =  0.9668737060041408
sum of Test accuracy =  [0.9723947550034506, 0.9668737060041408]
```

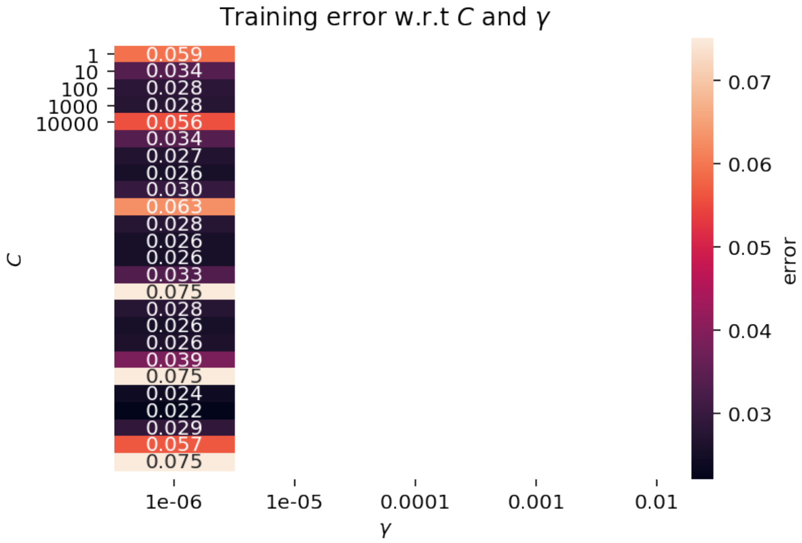


```
classifier: DecisionTree
BEST max depth D is 3
```

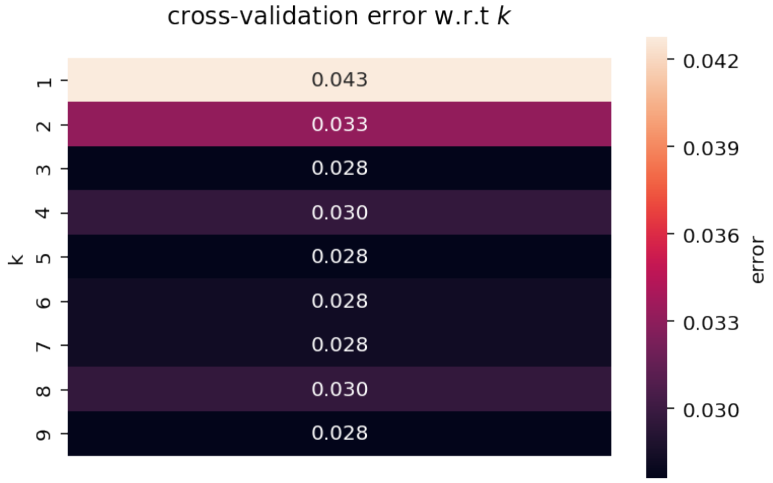
```
Training error =  0.013457647793877037
Training accuracy =  0.986542352206123
sum of training accuracy =  [0.9799864627651662, 0.986542352206123
]
Validation error =  0.01656126953824122
Validation accuracy =  0.9834387304617588
sum of Validation accuracy =  [0.979300799427276, 0.98343873046175
88]
Test error =  0.030365769496204287
Test accuracy =  0.9696342305037957
sum of Test accuracy =  [0.9772256728778468, 0.9696342305037957]
```



```
---------- trial: 3 ------------
seed: 17265

The shape of X_train is (1449, 8)
The shape of Y_train is (1449,)
The shape of X_test is (1449, 8)
The shape of Y_test is (1449,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-05
Training error =  0.017426586534170396
Training accuracy =  0.9825734134658296
sum of training accuracy =  [0.9822294784445569, 0.983264258724822
4, 0.9825734134658296]
Validation error =  0.02209760171 81721
Validation accuracy =  0.9779023982818279
sum of Validation accuracy =  [0.9806801097721035, 0.9827466889392
674, 0.9779023982818279]
Test error =  0.02208419599723943
Test accuracy =  0.9779158040027606
sum of Test accuracy =  [0.9786059351276742, 0.9682539682539683, 0
.9779158040027606]
```

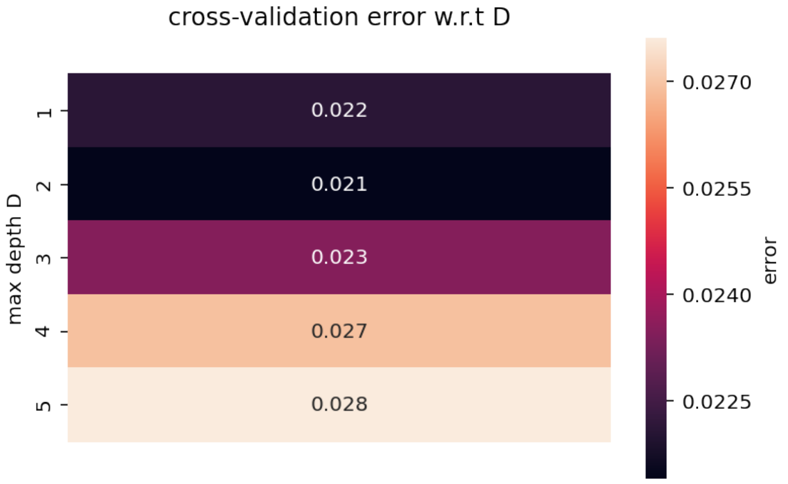


```
classifier: KNN
BEST_K IS 3
Training error =  0.02294769569486177
Training accuracy =  0.9770523043051382
sum of training accuracy =  [0.9756732914819553, 0.981193805599357
4, 0.9770523043051382]
Validation error =  0.027617229447560065
Validation accuracy =  0.9723827705524399
sum of Validation accuracy =  [0.9730915165254743, 0.9813673785944
397, 0.9723827705524399]
Test error =  0.029675638371290503
Test accuracy =  0.9703243616287095
sum of Test accuracy =  [0.9723947550034506, 0.9668737060041408, 0
.9703243616287095]
```



```
classifier: DecisionTree
BEST_max_depth D is 2
Training error =  0.018979203237035613
Training accuracy =  0.9810207967629644
sum of training accuracy =  [0.9799864627651662, 0.986542352206123
, 0.9810207967629644]
Validation error =  0.021403173845603307
Validation accuracy =  0.9785968261543967
```

```
sum of Validation accuracy =  [0.979300799427276, 0.98343873046175
88, 0.9785968261543967]
Test error =  0.022084195997239943
Test accuracy =  0.9779158040027606
sum of Test accuracy =  [0.9772256728778468, 0.9696342305037957, 0
.9779158040027606]
```



```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.9826890502117362 0.974
925235794801 0.9804430656643995
sum_svm= 2.9380573516709365
avg_train_knn, avg_test_knn, avg_vali_knn = 0.9779731337954836 0.9
698642742121003 0.9756138885574513
sum_knn = 2.923451296565035
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9825165372447512 0.9749
25235794801 0.9804454520144773
sum_dt = 2.9378872250540295
```

second dataset

In [26]:

```
#read in the dataset
cardio = pd.read_csv('cardio.csv')
print(cardio.isnull().sum())
cardio = cardio[:-65000]
X_and_Y = np.array(cardio).astype(np.float)
#X_and_Y = np.array(data).astype(np.float) # Load data from file.
np.random.seed(0)                # Set the random seed.
np.random.shuffle(X_and_Y)       # Shuffle the data.
X = X_and_Y[:, 0:-1]# First column to second last column: Features.
Y = X_and_Y[:, -1]               # Last column: Labels.
Y[Y==0] = -1                     # Convert labels from {0, 1} to {-1
, 1}.
size = len(X)
print(X.shape)
print(Y.shape)
print(X_and_Y[0])
```

```
id             0
age            0
gender         0
height         0
```

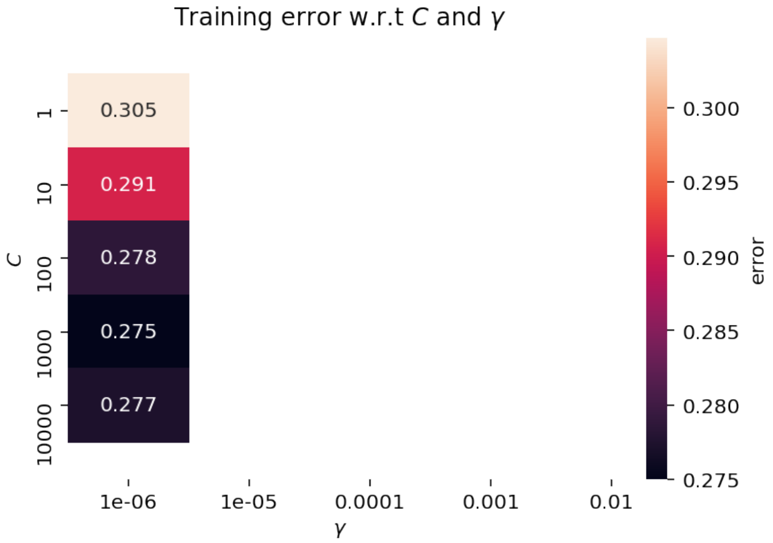
```
weight         0
ap_hi          0
ap_lo          0
cholesterol    0
gluc           0
smoke          0
alco           0
active         0
cardio         0
dtype: int64
(5000, 12)
(5000,)
[ 5.6600e+02  1.4492e+04  1.0000e+00  1.6300e+02  8.2000e+01  1.10
00e+02
  7.0000e+01  1.0000e+00  1.0000e+00  0.0000e+00  0.0000e+00  1.00
00e+00
 -1.0000e+00]
```

80 train 20 test

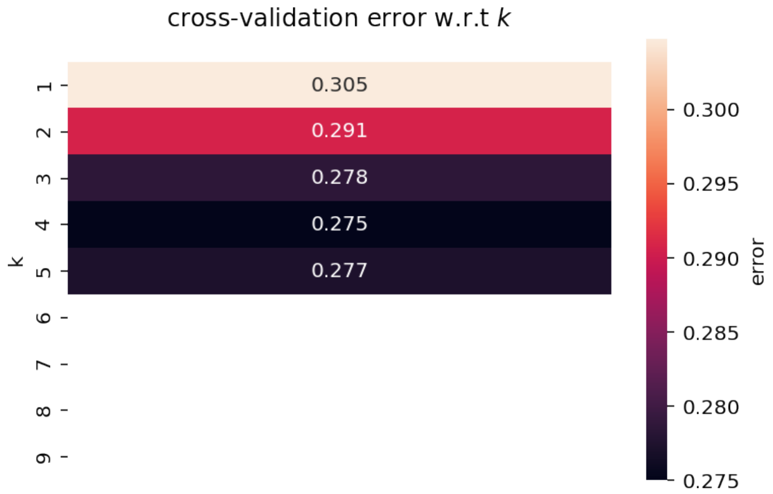
In [16]:

```
#80/20
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                   # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]                 # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]                  # Last column: Labels.
    Y[Y==0] = -1                        # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.2)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)


            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)


        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)
            draw_heatmap_SVM(opt[0], gamma_list, C_list)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validatio
n error w.r.t $k$')

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            print('Test accuracy = ', 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, avg_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn, avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)

avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

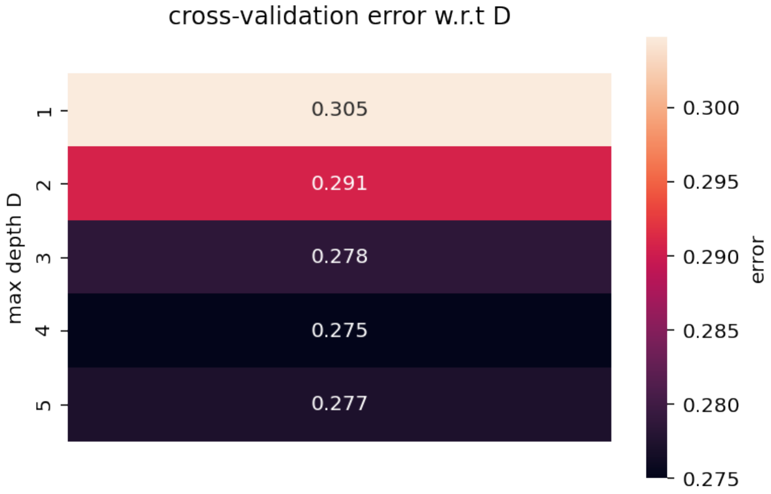
```
---------- trial: 1 ------------
seed: 15752

The shape of X_train is (4000, 12)
The shape of Y_train is (4000,)
The shape of X_test is (1000, 12)
The shape of Y_test is (1000,)

classifier: SVM
BEST_C IS 100
BEST_GAMMA IS 1e-06
Training error =  0.2574375
Training accuracy =  0.7425625
sum of training accuracy =  [0.7425625]
Validation error =  0.3265
Validation accuracy =  0.6735
sum of Validation accuracy =  [0.6735]
Test error =  0.33499999999999996
Test accuracy =  0.665
sum of Test accuracy =  [0.6735]

classifier: KNN
BEST_K IS 7
Training error =  0.30956249999999996
Training accuracy =  0.6904375
sum of training accuracy =  [0.6904375]
Validation error =  0.43700000000000006
Validation accuracy =  0.563
sum of Validation accuracy =  [0.563]
Test error =  0.44699999999999995
Test error =  0.44699999999999995
Test accuracy =  0.553
sum of Test accuracy =  [0.553]

classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.26312499999999994
Training accuracy =  0.7368750000000001
sum of training accuracy =  [0.7368750000000001]
```



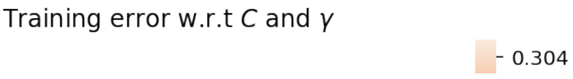
```
Validation error =  0.275
Validation accuracy =  0.725
sum of Validation accuracy =  [0.725]
```

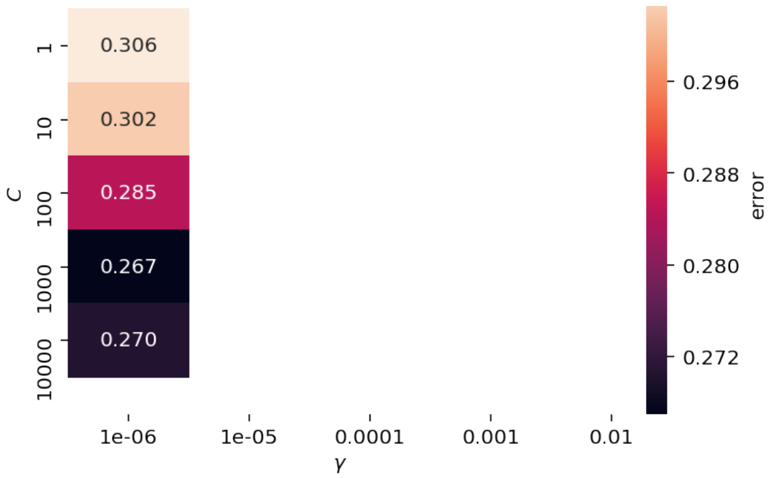


```
Test error =  0.30400000000000005
Test accuracy =  0.696
Test accuracy =  0.696
sum of Test accuracy =  [0.696]
```



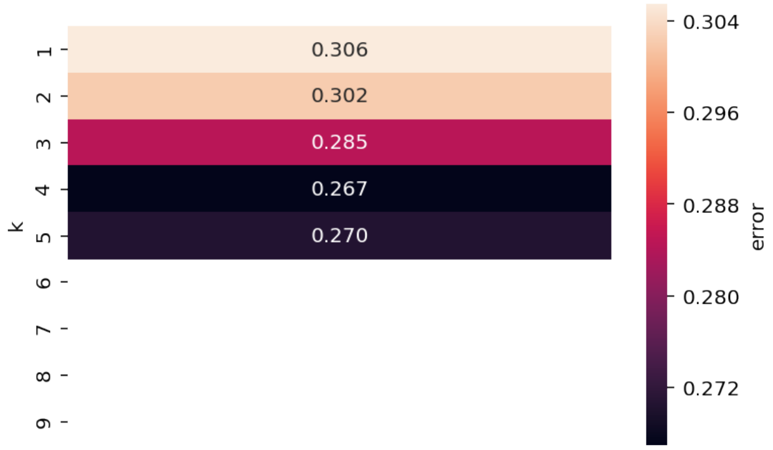
```
---------- trial: 2 ------------
seed: 13029

The shape of X_train is (4000, 12)
The shape of Y_train is (4000,)
The shape of X_test is (1000, 12)
The shape of Y_test is (1000,)

classifier: SVM
BEST_C IS 100
BEST_GAMMA IS 1e-06
Training error =  0.25325
Training accuracy =  0.74675
sum of training accuracy =  [0.7425625, 0.74675]
Validation error =  0.32575
Validation accuracy =  0.67425
sum of Validation accuracy =  [0.6735, 0.67425]
Test error =  0.32199999999999995
Test accuracy =  0.678
sum of Test accuracy =  [0.6735, 0.67425]

classifier: KNN
BEST_K IS 7
Training error =  0.3120624999999999
Training accuracy =  0.6879375000000001
sum of training accuracy =  [0.6904375, 0.6879375000000001]
Validation error =  0.43099999999999994
Validation accuracy =  0.5690000000000001
sum of Validation accuracy =  [0.563, 0.5690000000000001]
Test error =  0.42200000000000004
Test accuracy =  0.578
sum of Test accuracy =  [0.553, 0.578]

classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.261625
Training accuracy =  0.738375
sum of training accuracy =  [0.7368750000000001, 0.738375]
```





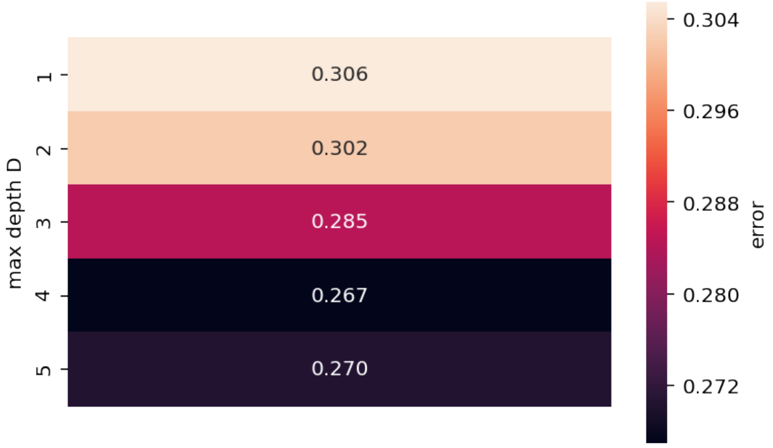
```
Validation error =  0.267
Validation accuracy =  0.733
sum of Validation accuracy =  [0.725, 0.733]
```

cross-validation error w.r.t *k*

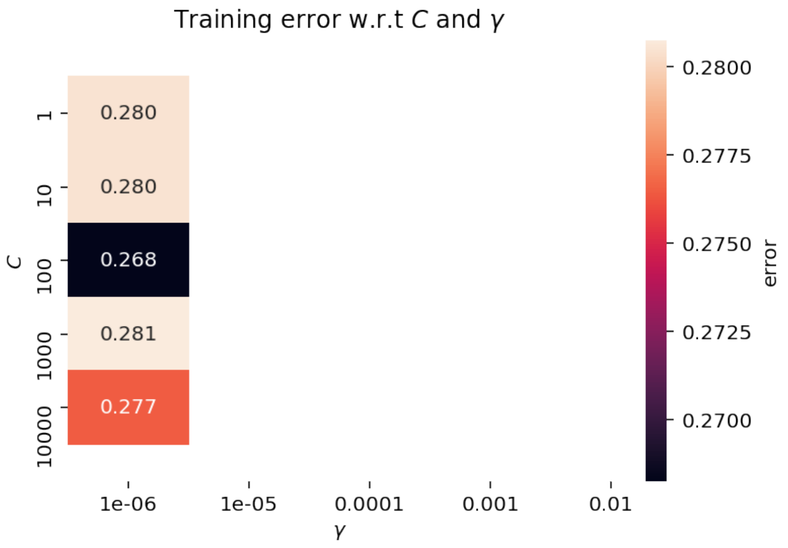


```
Test error =  0.28700000000000003
Test accuracy =  0.713
sum of Test accuracy =  [0.696, 0.713]
```

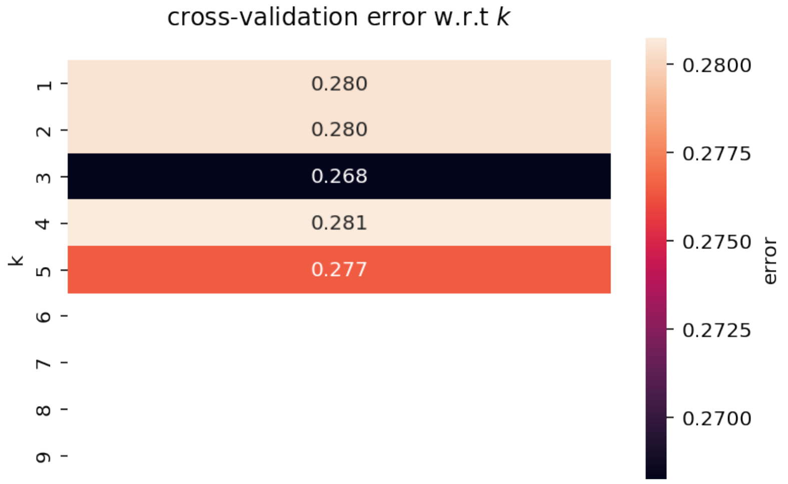
cross-validation error w.r.t D



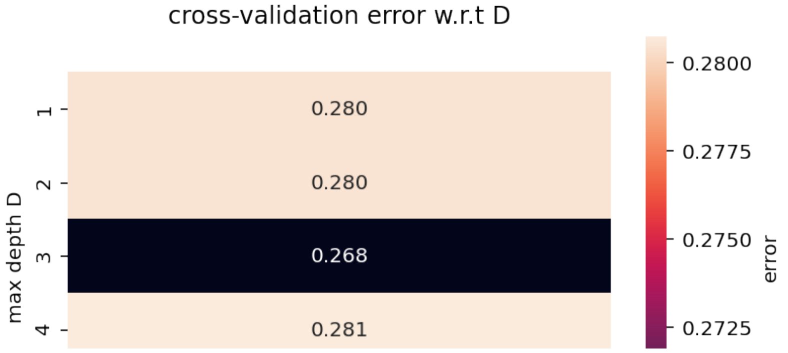
```
---------- trial: 3 ------------
seed: 37502

The shape of X_train is (4000, 12)
The shape of Y_train is (4000,)
The shape of X_test is (1000, 12)
The shape of Y_test is (1000,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.20768750000000002
Training accuracy =  0.7923125
sum of training accuracy =  [0.7425625, 0.74675, 0.7923125]
Validation error =  0.32099999999999995
Validation accuracy =  0.679
sum of Validation accuracy =  [0.6735, 0.67425, 0.679]
Test error =  0.345
Test accuracy =  0.655
sum of Test accuracy =  [0.6735, 0.67425, 0.679]

classifier: KNN
BEST_K IS 9
Training error =  0.33106250000000004
Training accuracy =  0.6689375
sum of training accuracy =  [0.6904375, 0.6879375000000001, 0.6689
375]
Validation error =  0.42825
Validation accuracy =  0.57175
sum of Validation accuracy =  [0.563, 0.5690000000000001, 0.57175]
Test error =  0.45499999999999996
Test accuracy =  0.545
sum of Test accuracy =  [0.553, 0.578, 0.545]

classifier: DecisionTree
BEST_max_depth D is 3
Training error =  0.2680625000000001
Training accuracy =  0.7319374999999999
sum of training accuracy =  [0.7368750000000001, 0.738375, 0.73193
74999999999]
```

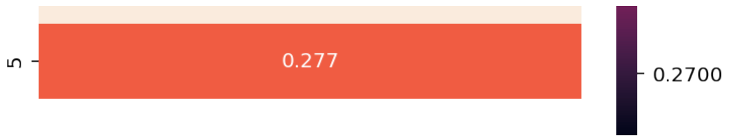


```
Validation error =  0.26825
Validation accuracy =  0.73175
sum of Validation accuracy =  [0.725, 0.733, 0.73175]
```



```
Test error =  0.30000000000000004
Test accuracy =  0.7
sum of Test accuracy =  [0.696, 0.713, 0.7]
```





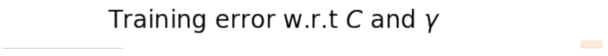
```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.7605416666666667 0.675
5833333333333 0.6755833333333333
sum_svm= 2.1117083333333335
avg_train_knn, avg_test_knn, avg_vali_knn = 0.6824375000000001 0.5
586666666666668 0.5679166666666667
sum_knn = 1.8090208333333337
avg_train_dt, avg_test_dt,avg_vali_dt =  0.7357291666666667 0.703
0.7299166666666667
sum_dt = 2.168645833333333
```

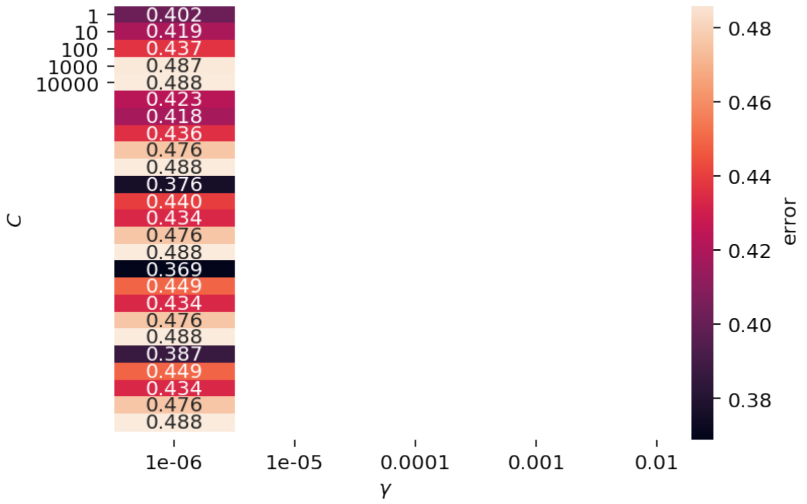
20 train 80 test

In [17]:

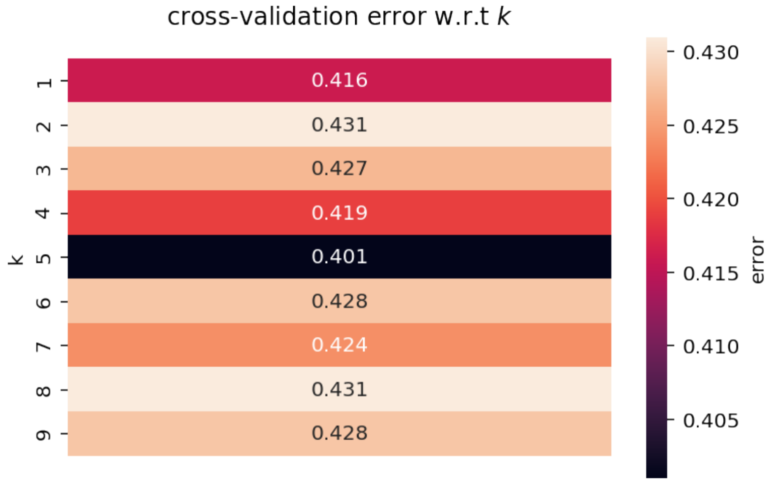
```
#20/80
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                 # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]               # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]                # Last column: Labels.
    Y[Y==0] = -1                      # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.8)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validatio
n error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

```
---------- trial: 1 ------------
seed: 5677

The shape of X_train is (1000, 12)
The shape of Y_train is (1000,)
The shape of X_test is (4000, 12)
The shape of Y_test is (4000,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.10675000000000012
Training accuracy =  0.8932499999999999
sum of training accuracy =  [0.8932499999999999]
Validation error =  0.369
Validation accuracy =  0.631
sum of Validation accuracy =  [0.631]
Test error =  0.38925
Test accuracy =  0.61075
sum of Test accuracy =  [0.631]
```

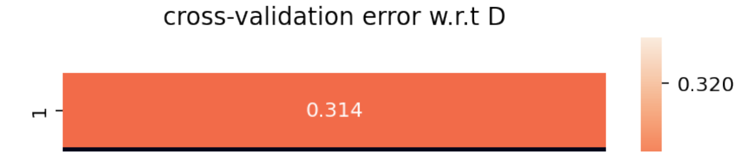


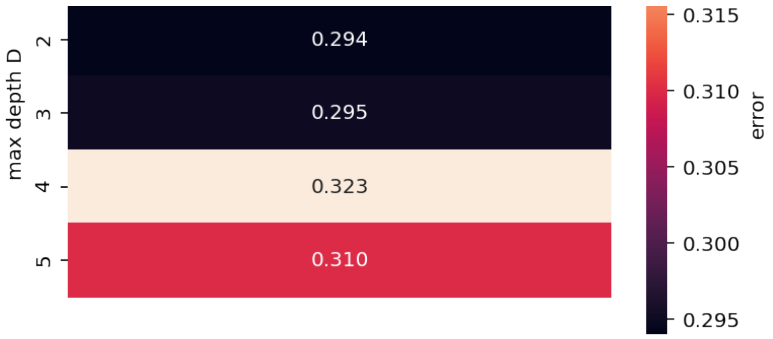


```
classifier: KNN
BEST_K IS 5
Training error =  0.27224999999999999
Training accuracy =  0.7277500000000001
sum of training accuracy =  [0.7277500000000001]
Validation error =  0.401
Validation accuracy =  0.599
sum of Validation accuracy =  [0.599]
Test error =  0.44475
Test accuracy =  0.55525
sum of Test accuracy =  [0.55525]
```

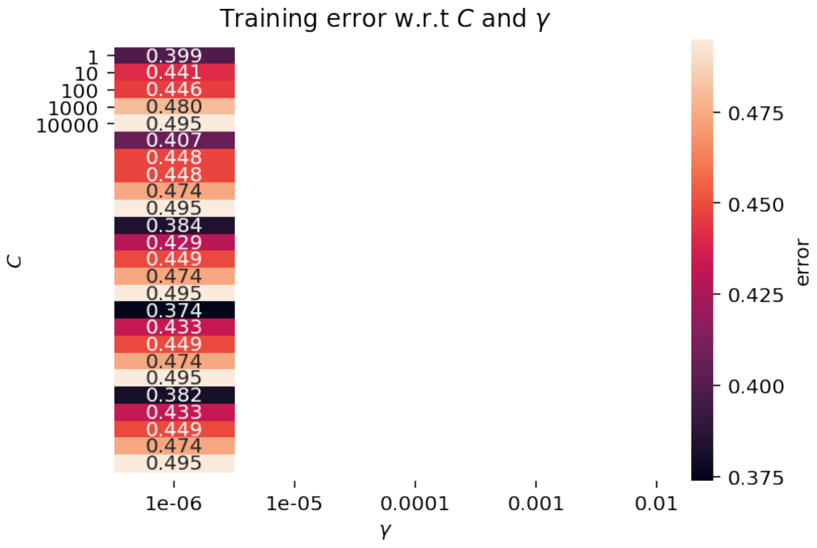


```
classifier: DecisionTree
BEST_max_depth D is 2
Training error =  0.28800000000000003
Training accuracy =  0.712
sum of training accuracy =  [0.712]
Validation error =  0.29399999999999993
Validation accuracy =  0.7060000000000001
sum of Validation accuracy =  [0.7060000000000001]
Test error =  0.3175
Test accuracy =  0.6825
sum of Test accuracy =  [0.6825]
```

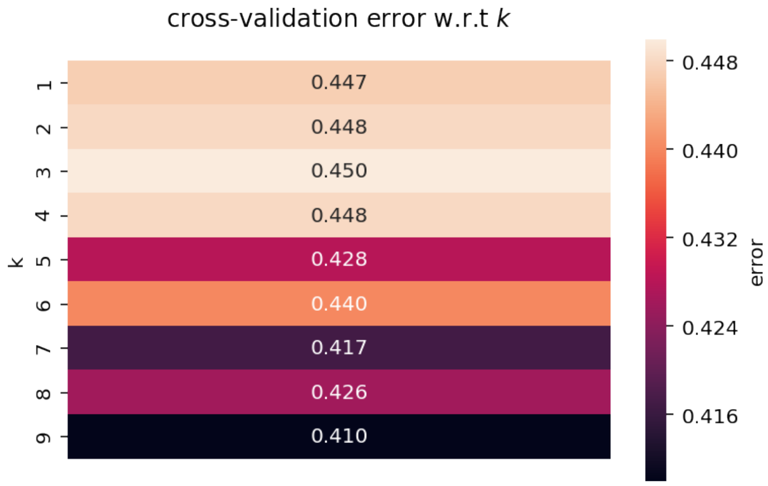




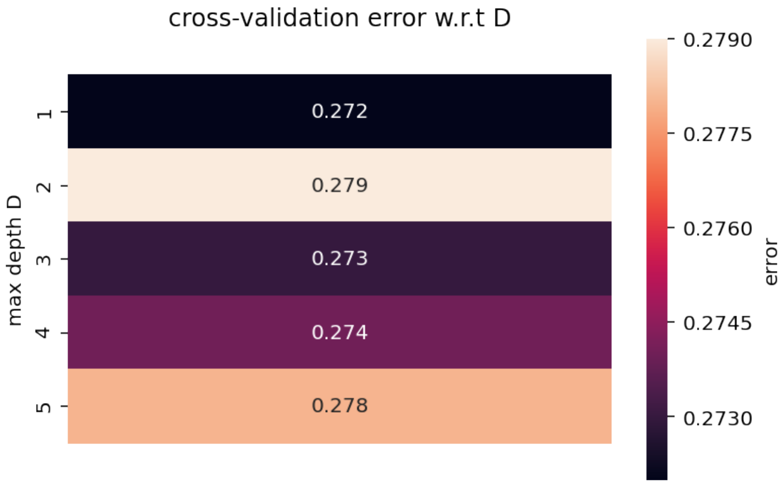
```
---------- trial: 2 ------------
seed: 49163

The shape of X_train is (1000, 12)
The shape of Y_train is (1000,)
The shape of X_test is (4000, 12)
The shape of Y_test is (4000,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.12374999999999992
Training accuracy =  0.8762500000000001
sum of training accuracy =  [0.8932499999999999, 0.876250000000000
1]
Validation error =  0.374
Validation accuracy =  0.626
sum of Validation accuracy =  [0.631, 0.626]
Test error =  0.36424999999999996
Test accuracy =  0.63575
sum of Test accuracy =  [0.631, 0.626]
```



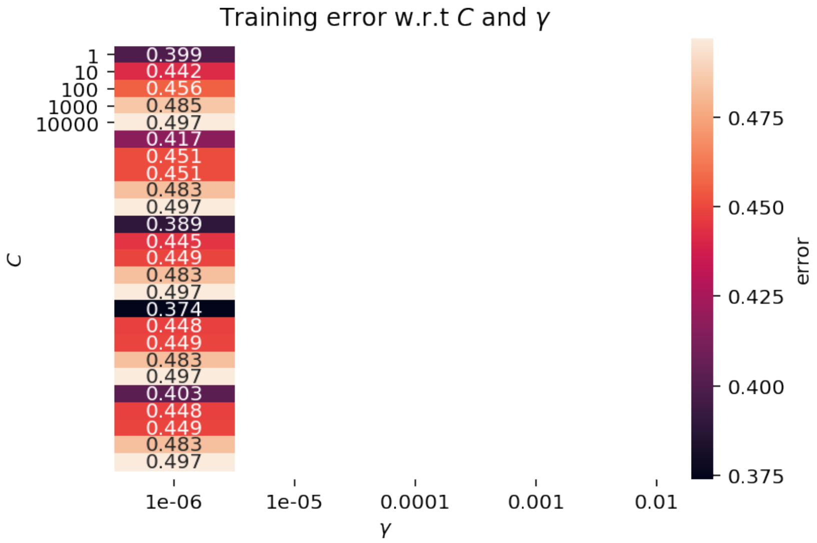
```
classifier: KNN
BEST_K IS 9
Training error =  0.32275
Training accuracy =  0.67725
sum of training accuracy =  [0.7277500000000001, 0.67725]
Validation error =  0.40999999999999999
Validation accuracy =  0.5900000000000001
sum of Validation accuracy =  [0.599, 0.5900000000000001]
sum of Validation accuracy =  [0.599, 0.5900000000000001]
Test error =  0.44925000000000004
Test accuracy =  0.55075
sum of Test accuracy =  [0.55525, 0.55075]
```



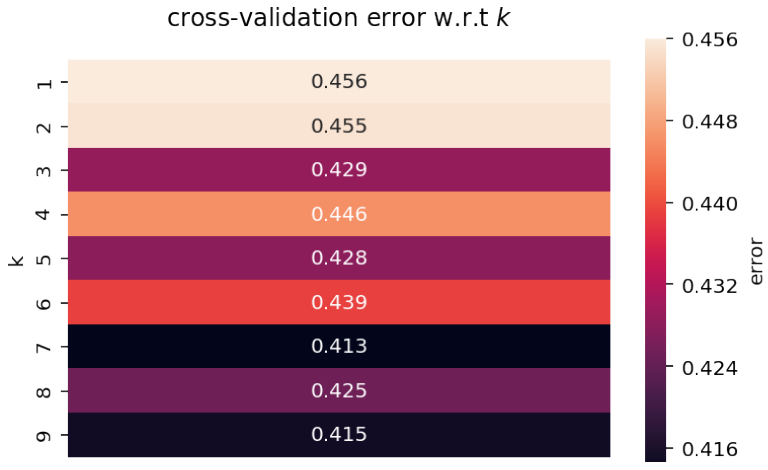
```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.272
Training accuracy =  0.728
sum of training accuracy =  [0.712, 0.728]
Validation error =  0.272
Validation accuracy =  0.728
sum of Validation accuracy =  [0.7060000000000001, 0.728]
Test error =  0.2905
Test accuracy =  0.7095
sum of Test accuracy =  [0.6825, 0.7095]
```



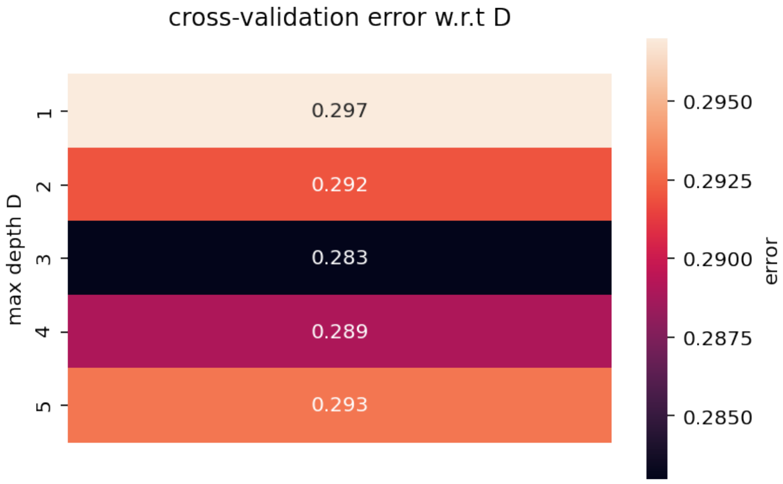
```
---------- trial: 3 ------------
seed: 46103

The shape of X_train is (1000, 12)
The shape of Y_train is (1000,)
The shape of X_test is (4000, 12)
The shape of Y_test is (4000,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
BEST_GAMMA IS 1e-06
Training error =  0.10399999999999987
Training accuracy =  0.8960000000000001
sum of training accuracy =  [0.8932499999999999, 0.8762500000000000
1, 0.8960000000000001]
Validation error =  0.374
Validation accuracy =  0.626
sum of Validation accuracy =  [0.631, 0.626, 0.626]
Test error =  0.37
Test accuracy =  0.63
sum of Test accuracy =  [0.631, 0.626, 0.626]
```



```
classifier: KNN
BEST_K IS 7
Training error =  0.31499999999999995
Training accuracy =  0.685
sum of training accuracy =  [0.7277500000000001, 0.67725, 0.685]
Validation error =  0.41300000000000003
Validation accuracy =  0.587
sum of Validation accuracy =  [0.599, 0.5900000000000001, 0.587]
Test error =  0.43574999999999997
Test accuracy =  0.56425
sum of Test accuracy =  [0.55525, 0.55075, 0.56425]
```



```
classifier: DecisionTree
BEST_max_depth D is 3
Training error =  0.24250000000000005
Training accuracy =  0.7575
sum of training accuracy =  [0.712, 0.728, 0.7575]
Validation error =  0.28299999999999999
Validation accuracy =  0.7170000000000001
sum of Validation accuracy =  [0.7060000000000001, 0.728, 0.717000
0000000001]
Test error =  0.3025
Test accuracy =  0.6975
sum of Test accuracy =  [0.6825, 0.7095, 0.6975]
```



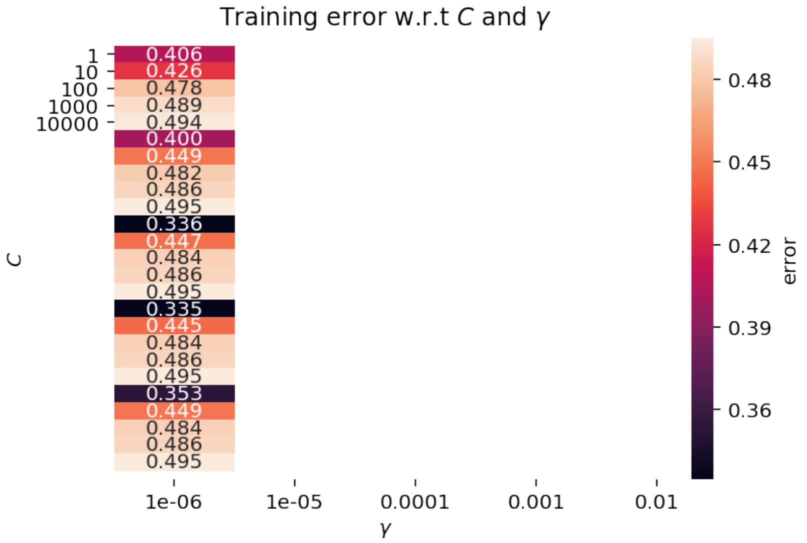
```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.8885 0.627666666666666
7 0.6276666666666667
sum_svm= 2.1438333333333333
avg_train_knn, avg_test_knn, avg_vali_knn = 0.6966666666666668 0.5
5675 0.592
sum_knn = 1.845416666666667
avg_train_dt, avg_test_dt,avg_vali_dt =  0.7324999999999999 0.6965
0.7170000000000001
sum_dt = 2.146
```

50 train 50 test

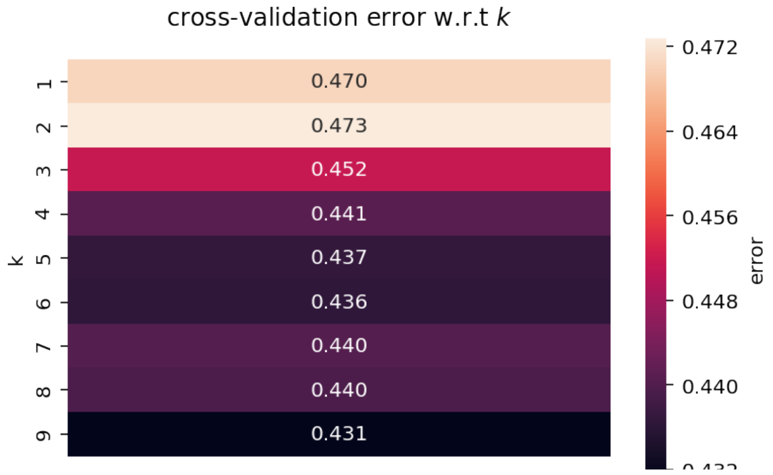
In [18]:

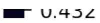
```
#50/50
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                   # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]                 # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]                  # Last column: Labels.
    Y[Y==0] = -1                        # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.5)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)
            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validatio
n error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)

sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

```
---------- trial: 1 ------------
seed: 38988

The shape of X_train is (2500, 12)
The shape of Y_train is (2500,)
The shape of X_test is (2500, 12)
The shape of Y_test is (2500,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.18290000000000006
Training accuracy =  0.8170999999999999
sum of training accuracy =  [0.8170999999999999]
Validation error =  0.3348
Validation accuracy =  0.6652
sum of Validation accuracy =  [0.6652]
Test error =  0.3456
Test accuracy =  0.6544
sum of Test accuracy =  [0.6652]
```

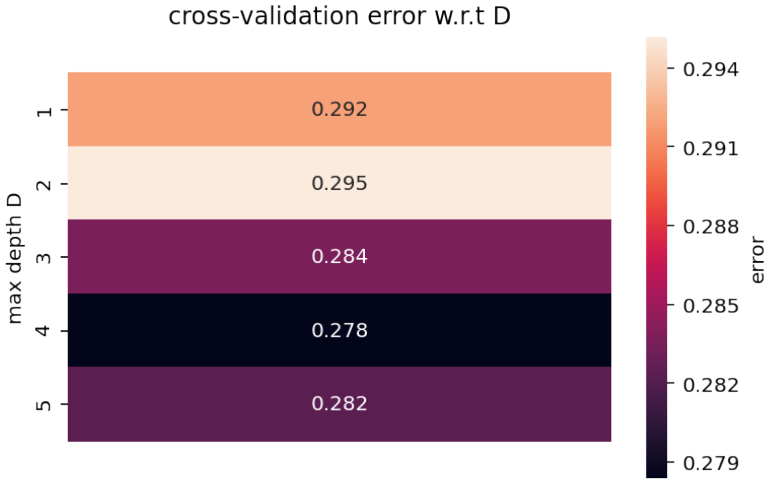


```
classifier: KNN
BEST_K IS 9
Training error =  0.3268000000000001
Training accuracy =  0.6731999999999999
sum of training accuracy =  [0.6731999999999999]
Validation error =  0.4312
Validation accuracy =  0.5688
sum of Validation accuracy =  [0.5688]
Test error =  0.43320000000000003
Test accuracy =  0.5668
sum of Test accuracy =  [0.5668]
```

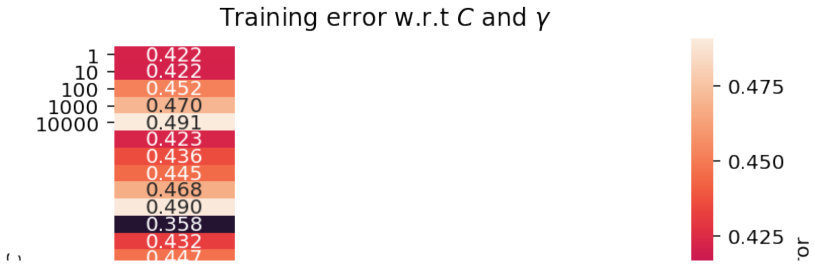


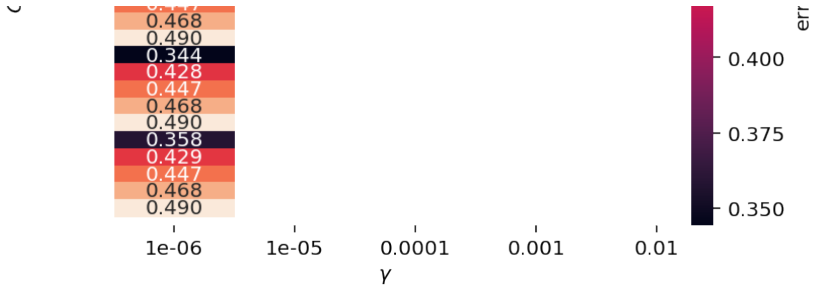


```
classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.2529
Training accuracy =  0.7471
sum of training accuracy =  [0.7471]
Validation error =  0.2784
Validation accuracy =  0.7216
sum of Validation accuracy =  [0.7216]
Test error =  0.29359999999999997
Test accuracy =  0.7064
sum of Test accuracy =  [0.7064]
```

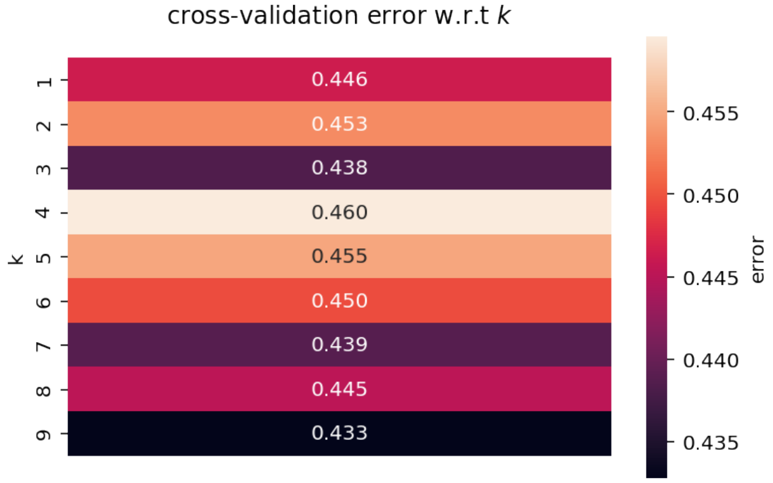

cross-validation error w.r.t D

```
---------- trial: 2 ------------
seed: 34328

The shape of X_train is (2500, 12)
The shape of Y_train is (2500,)
The shape of X_test is (2500, 12)
The shape of Y_test is (2500,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.19389999999999996
Training accuracy =  0.8061
sum of training accuracy =  [0.8170999999999999, 0.8061]
Validation error =  0.34440000000000004
Validation accuracy =  0.6556
sum of Validation accuracy =  [0.6652, 0.6556]
Test error =  0.33440000000000003
Test accuracy =  0.6656
sum of Test accuracy =  [0.6652, 0.6556]
```

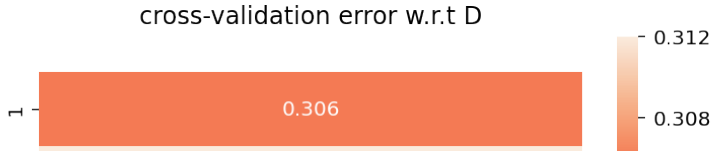

Training error w.r.t C and γ

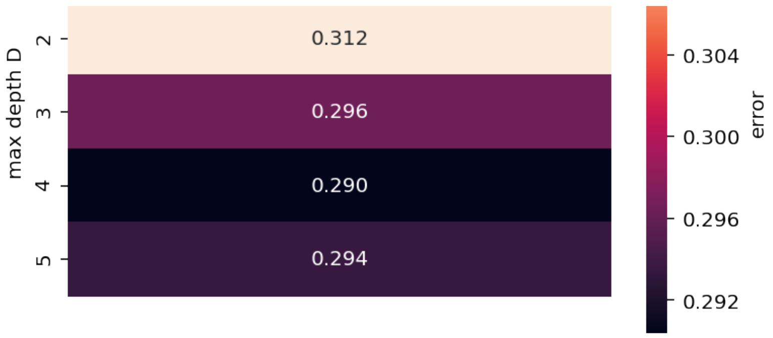


```
classifier: KNN
BEST_K IS 9
Training error =  0.33419999999999994
Training accuracy =  0.6658000000000001
sum of training accuracy =  [0.6731999999999999, 0.665800000000000
1]
Validation error =  0.4328000000000001
Validation accuracy =  0.5671999999999999
sum of Validation accuracy =  [0.5688, 0.5671999999999999]
Test error =  0.43720000000000003
Test accuracy =  0.5628
sum of Test accuracy =  [0.5668, 0.5628]
```

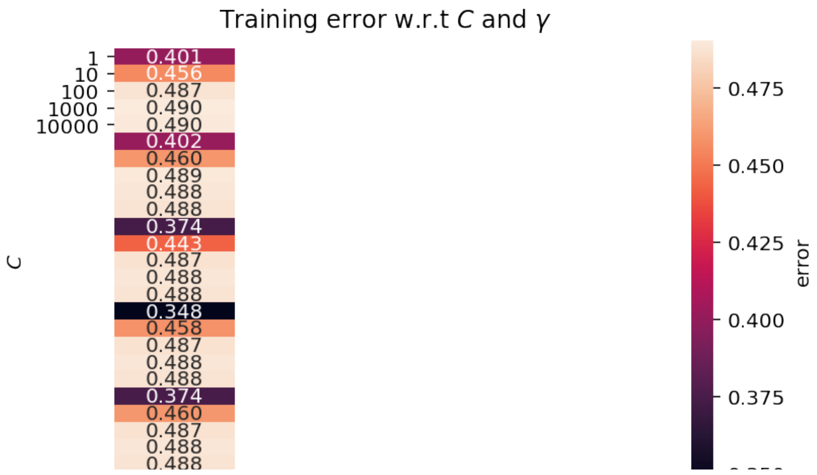


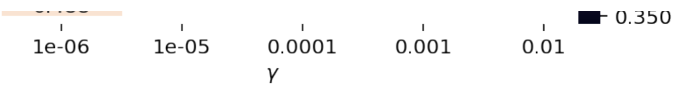
```
classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.26669999999999994
Training accuracy =  0.7333000000000001
sum of training accuracy =  [0.7471, 0.7333000000000001]
Validation error =  0.2904
Validation accuracy =  0.7096
sum of Validation accuracy =  [0.7216, 0.7096]
Test error =  0.28400000000000003
Test accuracy =  0.716
sum of Test accuracy =  [0.7064, 0.716]
```



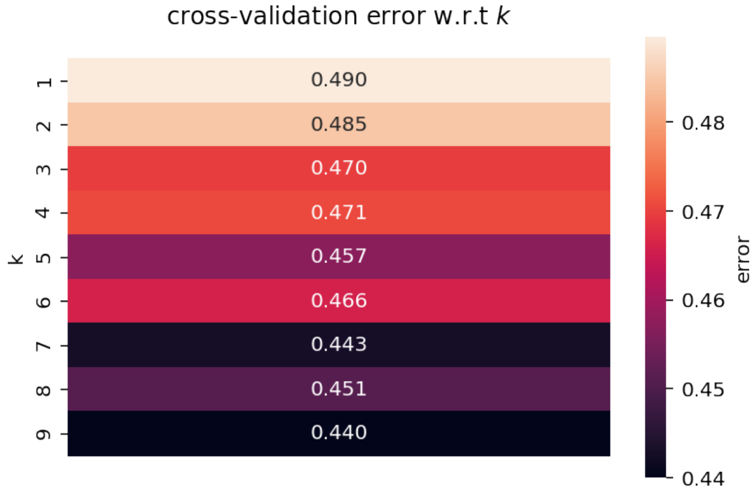


```
---------- trial: 3 ------------
seed: 47576

The shape of X_train is (2500, 12)
The shape of Y_train is (2500,)
The shape of X_test is (2500, 12)
The shape of Y_test is (2500,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.19169999999999998
Training accuracy =  0.8083
sum of training accuracy =  [0.8170999999999999, 0.8061, 0.8083]
Validation error =  0.34759999999999999
Validation accuracy =  0.6524000000000001
sum of Validation accuracy =  [0.6652, 0.6556, 0.6524000000000001]
Test error =  0.33640000000000003
Test accuracy =  0.6636
sum of Test accuracy =  [0.6652, 0.6556, 0.6524000000000001]
```

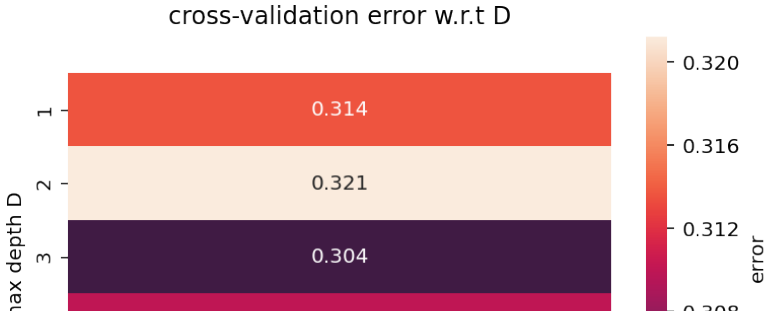


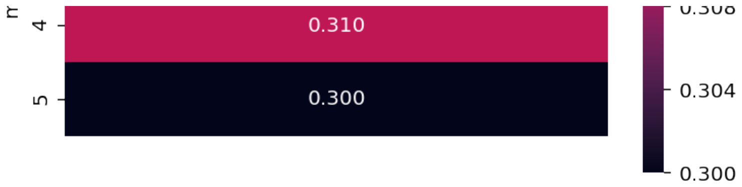


```
classifier: KNN
BEST_K IS 9
Training error =  0.3385
Training accuracy =  0.6615
sum of training accuracy =  [0.6731999999999999, 0.6658000000000000
1, 0.6615]
Validation error =  0.44000000000000006
Validation accuracy =  0.5599999999999999
sum of Validation accuracy =  [0.5688, 0.5671999999999999, 0.55999
99999999999]
Test error =  0.4424
Test accuracy =  0.5576
sum of Test accuracy =  [0.5668, 0.5628, 0.5576]
```



```
classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.26079999999999999
Training accuracy =  0.7392000000000001
sum of training accuracy =  [0.7471, 0.7333000000000001, 0.7392000
000000001]
Validation error =  0.30000000000000004
Validation accuracy =  0.7
sum of Validation accuracy =  [0.7216, 0.7096, 0.7]
Test error =  0.28159999999999996
Test accuracy =  0.7184
sum of Test accuracy =  [0.7064, 0.716, 0.7184]
```





```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.8104999999999999 0.657
7333333333334 0.6577333333333334
sum_svm= 2.125966666666667
avg_train_knn, avg_test_knn, avg_vali_knn = 0.6668333333333333 0.5
623999999999999 0.5653333333333332
sum_knn = 1.7945666666666664
avg_train_dt, avg_test_dt,avg_vali_dt =  0.7398666666666666 0.7136
0.7103999999999999
sum_dt = 2.1638666666666664
```

third dataset

In [28]:

```
bank = pd.read_csv('bank.csv')
print(bank.isnull().sum())
print(bank.head())

for i in bank.columns:
    print(bank[i].unique(),"\t",bank[i].nunique())

for i in bank.columns:
    print(bank[i].value_counts())
    print()

le=LabelEncoder()
for i in bank.columns:
    bank[i]=le.fit_transform(bank[i])

X_and_Y = np.array(bank).astype(np.float)
np.random.seed(0)                # Set the random seed.
np.random.shuffle(X_and_Y)       # Shuffle the data.
X = X_and_Y[:, 0:-1]# First column to second last column: Features.
Y = X_and_Y[:, -1]               # Last column: Labels.
Y[Y==0] = -1                     # Convert labels from {0, 1} to {-1
, 1}.
size = len(X)
print(X.shape)
print(Y.shape)
print(X_and_Y[0])
```

```
age             0
job             0
marital         0
education       0
default         0
balance         0
housing         0
loan            0
contact         0
```

```
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
   age          job  marital  education default  balance housing l
oan  \
0   30   unemployed  married    primary      no     1787      no
no
1   33     services  married  secondary      no     4789     yes
yes
2   35   management   single   tertiary      no     1350     yes

no
3   30   management  married   tertiary      no     1476     yes
yes
4   59  blue-collar  married  secondary      no        0     yes
no

    contact  day month  duration  campaign  pdays  previous poutco
me   y
0  cellular   19   oct        79         1     -1         0  unkno
wn  no
1  cellular   11   may       220         1    339         4  failu
re  no
2  cellular   16   apr       185         1    330         1  failu
re  no
3   unknown    3   jun       199         4     -1         0  unkno
wn  no
4   unknown    5   may       226         1     -1         0  unkno
wn  no
[30 33 35 59 36 39 41 43 20 31 40 56 37 25 38 42 44 26 55 67 53 68
32 49
 78 23 52 34 61 45 48 57 54 63 51 29 50 27 60 28 21 58 22 46 24 77
75 47
 70 65 64 62 66 19 81 83 80 71 72 69 79 73 86 74 76 87 84]       6
7
['unemployed' 'services' 'management' 'blue-collar' 'self-employed
'
 'technician' 'entrepreneur' 'admin.' 'student' 'housemaid' 'retir
ed'
 'unknown']      12
['married' 'single' 'divorced']         3
['primary' 'secondary' 'tertiary' 'unknown']     4
['no' 'yes']     2
[ 1787  4789  1350 ...  -333 -3313  1137]        2353
['no' 'yes']     2
['no' 'yes']     2
['cellular' 'unknown' 'telephone']       3
[19 11 16  3  5 23 14  6 17 20 13 30 29 27  7 18 12 21 26 22  2  4
15  8
 28  9  1 10 31 25 24]   31
['oct' 'may' 'apr' 'jun' 'feb' 'aug' 'jan' 'jul' 'nov' 'sep' 'mar'
'dec']   12
[  79  220  185  199  226  141  341  151   57  313  273  113  328
261
   89  189  239  114  250  148   96  140  109  125  169  182  247
```

```
119
  149   74  897   81   40  958  354  150   97  132  765   16  609
106
  365  205   11  105   59  425  204  181 1018 1740   98  441  272
159
  295  314  579  554  323  227  134  223  155  130  630  164  268
380
  154  221   67  367   87  701  652   63  398  224  406   60  521
279
  203  201  372  391  165  231  291  233  473  736  337  553  345
65
    9  259  371  280  243  435  258    7  317   76  170  386   83
69
  564  588  779  281 1877   51   32  176  161  187   24   85  236
54
   71  489   39  455   86  190   45  168  194  103  333  102   92
213
  289   77  324   84   10   35   82  676   80  549  135  412  101
253
  166   18  147   14   61  377  152  382  543  240   48  471  285
301
  768 1337  403  139  196  115   17   95  198  654  256  834   20
178
  111  186  297  210  112  222  195  123  145  124  216  483  690
344
  673  144  246  361  375 1097  180  373  230   58   88  487   29
484
  262  644  699   49   64  121  197  331  138  312  120  526  211
62
  988  451 1030 1484  445  383  605  330  171  442  772  249  357
271
  783  472  395   56  641  429  157  162  799 1370   22  215 1017
298
  126    8  555  270  339  342 1434   30  397  620    6  209  419
283
  188  267  245 1065  207  456  131   94  567  153   53  234  108
208
  597  505  332  212  493  681  287  202   37   72  325 1212  319
514
  551  142  293  107  127 1816  200  418  387  156   47  265   31
28
  369  854   46  266  321   99  430  264  118  343    5  722  748
523
  421   15  502  193  347  468  388 1735  172  117  587  501  282
110
  104  378 1407  738   70  904  336  238  585   68 1713  218  661
566
  136  160   44  792   73   90  346  192  682  651  405  350   36
389
 3025  219  427  533   19  819  278  617   34  668   75  146  356
251
  352  184  568  260  447  426  174  284  428  237 1031  700  590
43
   27 1181  122  307  770  767  232  986   66  158  306  559  183
631
 1282 1199  244   55  290  385  133   91   25  275  632  100   41
446
  304  335  276   42  614  557 1663  510 1259  225  404 1015  761
464
  206  667  143  717   38  254  882  957  299  167  500  177  457
```

```
460
 1028  315  381  643  508  128  492  257  241  536  601 1168  277
364
  229  402  175  255  820  116  463  603  191 2087  754  303  288
891
  558  228  353  296  432 1130  305  274  860  420  756  968  408
13
  763  316   50    4   78  286  766  648  688   21  593  407  563
52
  803  396  637  945 1178  506  409  327  618  936  329  179  731

670
  318  415  137  349  263  671  452  163  586  650  610  747  252
883
  684  686 1060  724  424  712  753 1081  376  433  411 1083  757
524
  653   93  503  217  475  340  242  530   23  935  773  423  626
578
  248  528  785  952 1174  915  937  129 1063  758  574  847 1558
789
 1441  322 1504  537  611   26   12  235  796 1126  697  931 1034
362
  410  570  633  659  302  727  214  173  635  540 1210  486  646
326
  414  716  449  580  399 1029  755  619  606  971  348  594 1275
379
 1032  393  808  923  413  541  602  762  360  310  311  638  355
300
  417  308  657  434  488 1309 1056  908  401  827  735  691  461
669
 1473 1386  294  910  550 1366 1532  955  513 1236  809  482 1164
674
  709  436  374  309  363  422  358  640  439  476  480  517  993
33
  730  636  750  334  868  351 1689  607  485 1021  732  577  733
788
  863 1073  525  696  535  370  465  338  956  546  470  836  544
443
 1149  707 1451 1143  477  450  467  292  806  560  759 1183  598
466
  431  269  542  562  515 1165  547  780  916  474  509  679 1472
965
 1139  504  672  749  454  531  479  384  973  728  656  998  320
615
  612  664  394  877 1971 1258  599  655 1994  743  924  929  491
719
  565 1529  390 1467 1007  665  990  582  458  775  497  698  702
520
 1156  884 1521  359  527  994 1579  437 1225  865  569 1011  814
984
  392  595 1448  529 1006  622  494 1022 1044  781 1124  400  516
1516
  978  720  495  798  876  875  481  793 1117 1223 1101  744 2769
561
  715  639  538 1721 1608  725  519  490  907  680  977  959  693
2029
 1009  718  805  623  976  600  469 1010  634 1531  764  532  825
539
  816  821 1231  742 2456  721  777  548  830  645  723  746  869
1173
  624  518  815  857  921  627  800  366 1088  812  866 1151  873
```

```
592
  663  576 1476  951 1234 1263  660]     875
[ 1  4  2  5  3  6 18 10  9  7 12 14 13 24 11  8 29 32 16 22 15 30
25 21
 17 19 23 20 50 28 31 44]        32
[ -1 339 330 176 147 241 152 105 342 101   5  92  56 170 182 297 1
96 460
 137 367 145 169 207 266 288 168 345 436  90 183 146 335 347 119
7 271
 181  88 141 126  61 373 351 242  62  91 308 250 172 265  78  28
79   1
 188 167  89 164 462 209 321 254  94 364  96 356 149 363 275 325 3
41 260
 358  87 303  98 327 337 322 102  99 370  84 212  63  81 191 360 3
32  80
  85 247 150 175 382 261 336  58 206 112 199 133 208 253 135 278 1
40 298
 273 124 281 162 323 349 117   2 256 333 116 268 136 198 357 259 3
53 174
 371 205 246  69 315 110 461 184 270 127 187  64 130 346 100 352 8
08 113
 378 292 287 107 293 139 138 193 274  97 103 359 185 674 211 300 3
34 280
 479  95 262 362 225   3 366  60 190 368 122 343 131 365 299 115 3
16 180
 154 313 264 350  73 232 204 143 375 186 344 210 248 177 221 189 1
04 258
 305 171 120 317 178 386 118 404 374 282 179 284 227 291 173 871 2
38 294
 222 435 340 426 239  83 111 415 255 235 244  38 683 329  59 151 1
92 158
 338 388 165 348 197 295 109 484 326 369 397 414 319 474  93 249 2
72 355
 195  82 541 231 153 201 761 114 385 267 161 467  75 106 223 312 1
48 309
 283  86 166 160 450 500 311 123 159 687 224 361  74  76 286  77
57 219
 331 804 144 234]        292
[ 0  4  1  3  2  5 20  7  6 10  9  8 18 19 12 13 11 14 15 24 17 22
23 25]   24
['unknown' 'failure' 'other' 'success']          4
['no' 'yes']     2
34    231
32    224
31    199
36    188
33    186
35    180
37    161
38    159
30    150
40    142
42    141
41    135
39    130
46    119
43    115
48    114
45    112
49    112
```

```
47    108
44    105
28    103
50    101
29     97
27     94
53     94
57     91
51     91
55     90
52     86
58     85
     ...
23     20
61     16
22      9
66      9
63      8
70      7
21      7
64      7
62      7
65      6
77      6
69      6
71      6
80      6
75      6
73      6
67      5
19      4
83      4
72      4
79      4
20      3
74      3
78      3
68      2
76      2
84      1
81      1
86      1
87      1
Name: age, Length: 67, dtype: int64

management       969
blue-collar      946
technician       768
admin.           478
services         417
retired          230
self-employed    183
entrepreneur     168
unemployed       128
housemaid        112
student           84
unknown           38
Name: job, dtype: int64

married     2797
```

```
single      1196

divorced     528
Name: marital, dtype: int64

secondary    2306
tertiary     1350
primary       678
unknown       187
Name: education, dtype: int64

no     4445
yes      76
Name: default, dtype: int64

 0        357
 2         24
 1         15
 4         13
 23        11
 5         11
 179       10
 6         10
 14        10
 35        10
 61         9
 137        9
 92         9
 50         9
 25         9
 60         9
 8          9
 174        9
 260        8
 3          8
 168        8
 171        8
 16         8
 225        7
 209        7
 473        7
 47         7
 33         7
 79         7
 13         7
         ...
-145        1
-2          1
 1553       1
 1557       1
 7702       1
 1465       1
 3496       1
 5431       1
 1397       1
 1337       1
 2226       1
 1361       1
 3412       1

 4265       1
 1372       1
 1373       1
 3432       1
 3436       1
 13683      1
 7546       1
 5543       1
 3452       1
 1405       1
 1409       1
 1413       1
 13711      1
 3472       1
 5527       1
 1433       1
 1445       1
 4094       1
Name: balance, Length: 2353, dtype: int64

yes    2559
no     1962
Name: housing, dtype: int64

no     3830
yes     691
Name: loan, dtype: int64

cellular     2896
unknown      1324
telephone     301
Name: contact, dtype: int64

20    257
18    226
19    201
21    198
14    195
17    191
7     190
6     187
28    181
5     181
8     180
29    175
15    174
30    168
13    166
16    164
9     163
11    152
12    151
4     139
2     114
27    113
26    110
3     105
23    102
22     86
25     80
31     59
10     50
```

```
24      36
1       27
Name: day, dtype: int64

may    1398
jul     706
aug     633
jun     531
nov     389
apr     293
feb     222
jan     148
oct      80
sep      52
mar      49
dec      20
Name: month, dtype: int64

123     27
104     25
119     23
106     22
77      22
121     22
58      22
71      20
168     20
161     19
147     19
151     19
113     19
127     19
89      19
76      19
97      19
185     18
101     18
144     18
112     18
107     18
159     18
85      17
205     17
118     17
143     17
74      17
82      17
172     17
        ..
682      1
686      1
690      1
1529     1
1337     1
877      1
1021     1
921      1
929      1
937      1
945      1
957      1
```

```
957      1
965      1
973      1
977      1
993      1
1009     1
1029     1
1309     1
1065     1
1073     1
1081     1
1097     1
1101     1
1117     1
1149     1
1165     1
1173     1
1225     1
4        1
Name: duration, Length: 875, dtype: int64

1     1734
2     1264
3      558
4      325
5      167
6      155
7       75
8       56
9       30
10      27
11      22
12      21
13      17
14      10
15       9
16       8
17       7
18       7
25       4
28       3
24       3
20       3
19       3
21       2
23       2
22       2
32       2
29       1
44       1
30       1
50       1
31       1
Name: campaign, dtype: int64

-1      3705
 182      23
 183      20
 363      12
 92       12
 91       11
 91       11
 169      10
 181      10
 370       9
 364       9
 349       8
 167       8
 99        8
 94        8
 184       7
 2         7
 176       7
 172       7
 175       6
 87        6
 345       6
 185       6
 95        6
 152       6
 173       6
 342       6
 96        6
 85        6
 150       6
 351       6
        ...
 541       1
 397       1
 385       1
 341       1
 329       1
 313       1
 305       1
 265       1
 225       1
 205       1
 687       1
 118       1
 386       1
 158       1
 382       1
 378       1
 374       1
 294       1
 282       1
 262       1
 254       1
 250       1
 242       1
 238       1
 234       1
 222       1
 210       1
 206       1
 162       1
 28        1
Name: pdays, Length: 292, dtype: int64

0     3705
1      286
```

```
2      193
3      113
4       78
5       47
6       25
7       22
8       18
9       10
12       5
10       4
11       3
14       2
20       1
24       1
23       1
13       1
17       1
19       1
18       1
22       1
15       1
25       1
Name: previous, dtype: int64

unknown    3705
failure     490
other       197
success     129
Name: poutcome, dtype: int64

no     4000
yes     521
Name: y, dtype: int64

(4521, 16)
(4521,)
[ 16.    1.    1.    0.    0. 638.    1.    0.    2.   15.    8. 421.    2.
0.
   0.    3.   -1.]
```

train 80 test 20

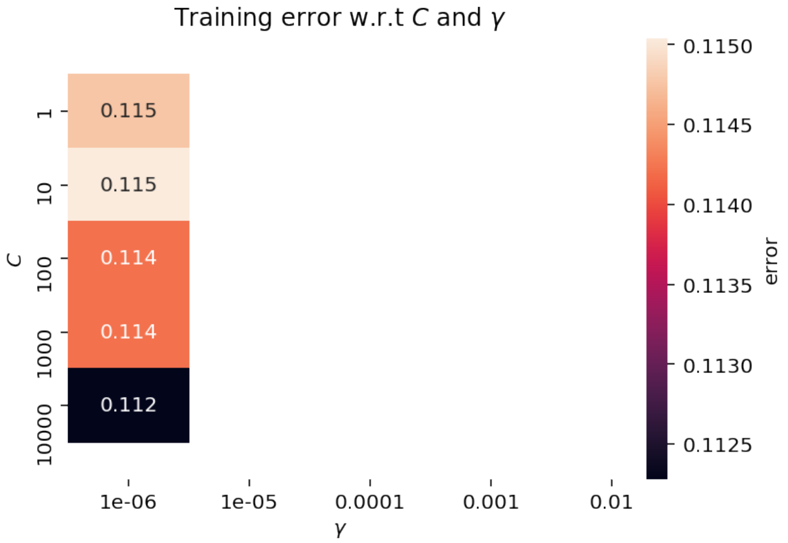
In [20]:

```
#80/20
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []
```

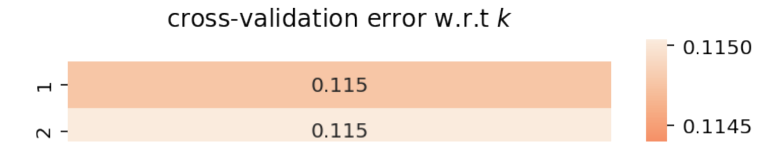
```python
for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)               # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]             # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]              # Last column: Labels.
    Y[Y==0] = -1                    # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.2)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)


        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)
        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)
            draw_heatmap_SVM(opt[0], gamma_list, C_list)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validatio
n error w.r.t $k$')

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)

avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

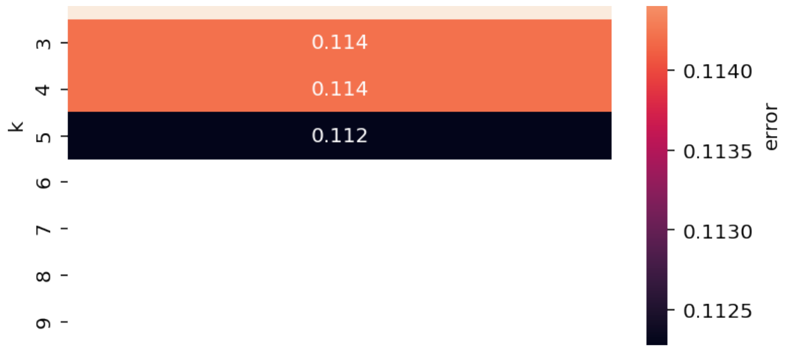
```
---------- trial: 1 ------------
seed: 42568

The shape of X_train is (3616, 16)
The shape of Y_train is (3616,)
The shape of X_test is (905, 16)
The shape of Y_test is (905,)
```

```
classifier: SVM
BEST_C IS 1
BEST_GAMMA IS 0.0001
Training error =  0.09036241435544079
Training accuracy =  0.9096375856445592
sum of training accuracy =  [0.9096375856445592]
Validation error =  0.11393900491353559
Validation accuracy =  0.8860609950864644
sum of Validation accuracy =  [0.8860609950864644]
Test error =  0.11270718232044197
Test accuracy =  0.887292817679558
sum of Test accuracy =  [0.8860609950864644]

classifier: KNN
BEST_K IS 8
Training error =  0.10508854539430568
Training accuracy =  0.8949114546056943
sum of training accuracy =  [0.8949114546056943]
Validation error =  0.11532174869902112
Validation accuracy =  0.8846782513009789
sum of Validation accuracy =  [0.8846782513009789]
Test error =  0.11270718232044197
Test accuracy =  0.887292817679558
sum of Test accuracy =  [0.887292817679558]

classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.09319705742721385
Training accuracy =  0.9068029425727862
sum of training accuracy =  [0.9068029425727862]
```

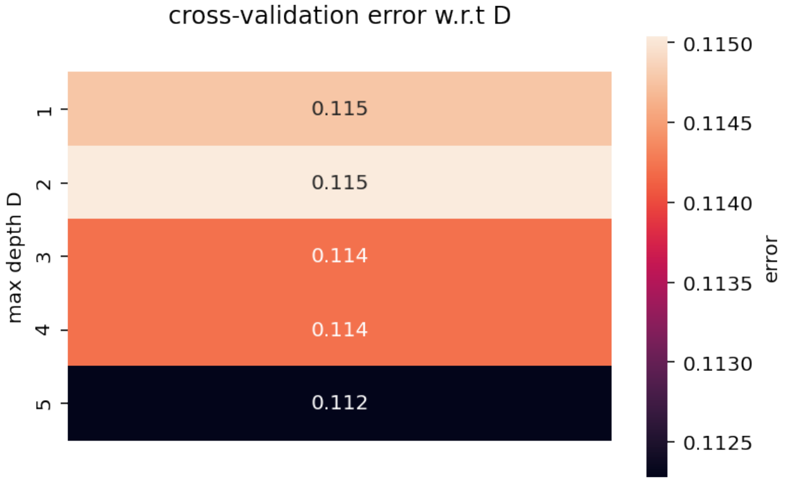


```
Validation error =  0.11228116427103152
Validation accuracy =  0.8877188357289685
sum of Validation accuracy =  [0.8877188357289685]
```

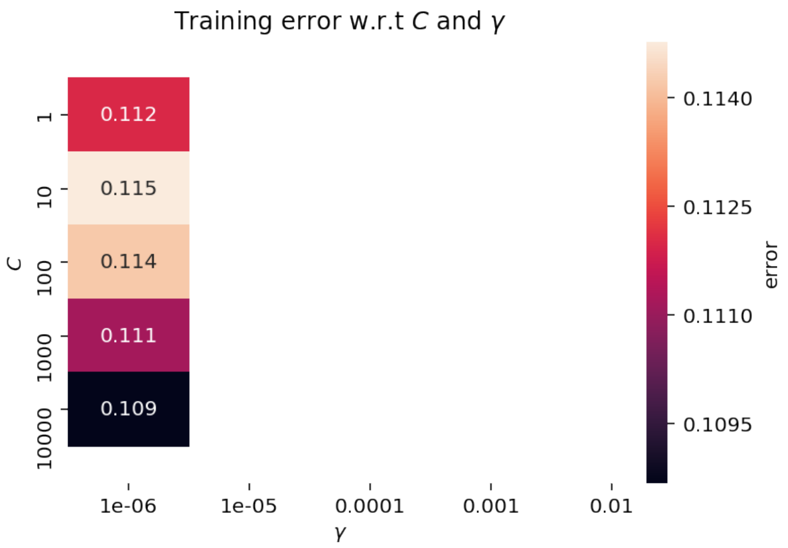




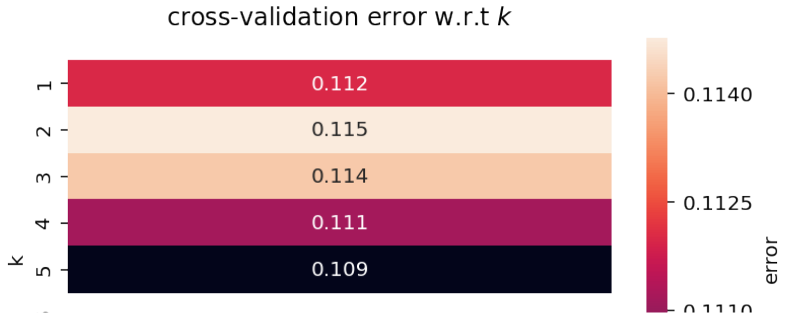
```
Test error =  0.11270718232044197
Test accuracy =  0.887292817679558
sum of Test accuracy =  [0.887292817679558]
```



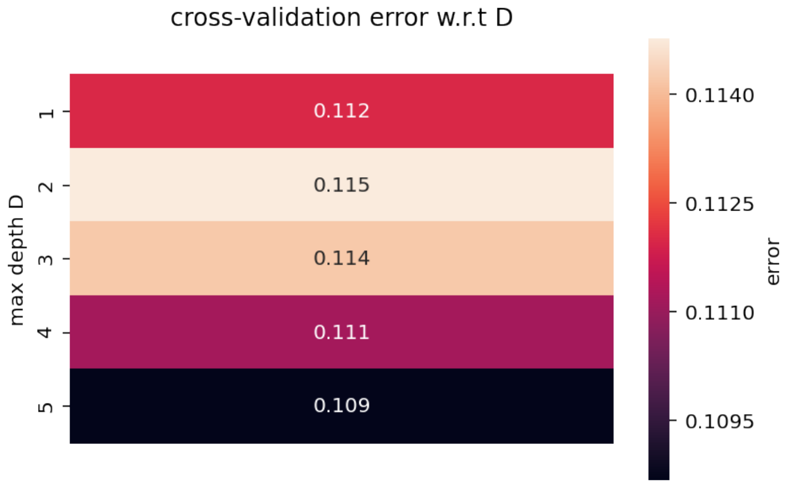
```
---------- trial: 2 ------------
seed: 44188

The shape of X_train is (3616, 16)
The shape of Y_train is (3616,)
The shape of X_test is (905, 16)
The shape of Y_test is (905,)

classifier: SVM
BEST_C IS 100
BEST_GAMMA IS 1e-05
Training error =  0.08261894141388648
Training accuracy =  0.9173810585861135
sum of training accuracy =  [0.9096375856445592, 0.917381058586113
5]
Validation error =  0.10840611937675282
Validation accuracy =  0.8915938806232472
sum of Validation accuracy =  [0.8860609950864644, 0.8915938806232
472]
Test error =  0.13370165745856355
Test accuracy =  0.8662983425414365
sum of Test accuracy =  [0.8860609950864644, 0.8915938806232472]

classifier: KNN
BEST_K IS 9
Training error =  0.10017959600103077
Training accuracy =  0.8998204039989692
sum of training accuracy =  [0.8949114546056943, 0.899820403998969
2]
Validation error =  0.11034211350801981
Validation accuracy =  0.8896578864919802
sum of Validation accuracy =  [0.8846782513009789, 0.8896578864919
802]
Test error =  0.1359116022099448
Test accuracy =  0.8640883977900552
sum of Test accuracy =  [0.887292817679558, 0.8640883977900552]

classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.0895325388367686
Training accuracy =  0.9104674611632314
sum of training accuracy =  [0.9068029425727862, 0.910467461163231
4]
```



```
Validation error =  0.10868274454964355
Validation accuracy =  0.8913172554503564
sum of Validation accuracy =  [0.8877188357289685, 0.8913172554503
564]
```

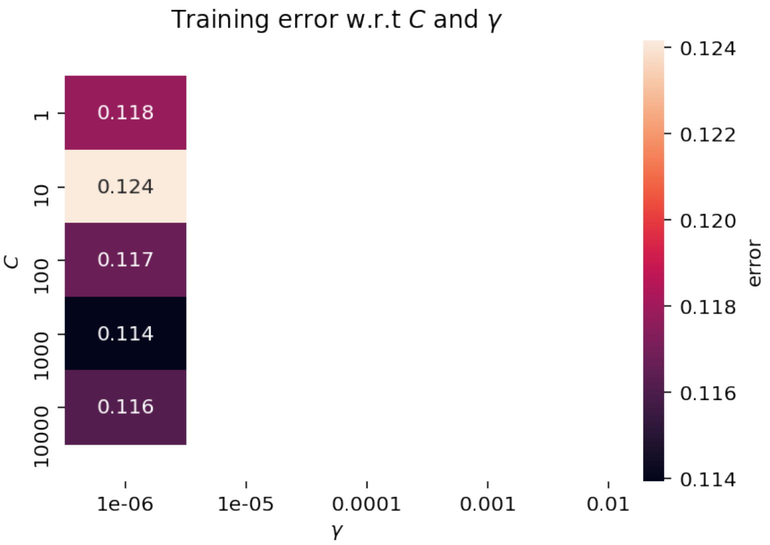


```
Test error =  0.10828729281767957
Test accuracy =  0.8917127071823204
sum of Test accuracy =  [0.887292817679558, 0.8917127071823204]
```

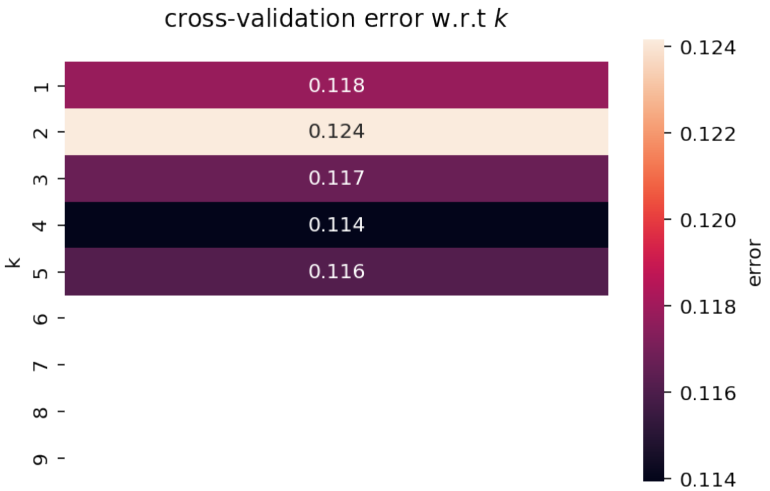


```
---------- trial: 3 ------------
seed: 34615

The shape of X_train is (3616, 16)
The shape of Y_train is (3616,)
The shape of X_test is (905, 16)
The shape of Y_test is (905,)

classifier: SVM
BEST_C IS 100
BEST_GAMMA IS 1e-06
Training error =  0.11165671992155435
Training accuracy =  0.8883432800784457
sum of training accuracy =  [0.9096375856445592, 0.917381058586113
5, 0.8883432800784457]
Validation error =  0.11587423488686643
Validation accuracy =  0.8841257651131336
sum of Validation accuracy =  [0.8860609950864644, 0.8915938806232
472, 0.8841257651131336]
Test error =  0.10607734806629832
Test accuracy =  0.8939226519337017
sum of Test accuracy =  [0.8860609950864644, 0.8915938806232472, 0
.8841257651131336]

classifier: KNN
BEST_K IS 8
Training error =  0.10598717082632325
Training accuracy =  0.8940128291736767
sum of training accuracy =  [0.8949114546056943, 0.899820403998969
2, 0.8940128291736767]
Validation error =  0.12085157760405918
Validation accuracy =  0.8791484223959408
sum of Validation accuracy =  [0.8846782513009789, 0.8896578864919
802, 0.8791484223959408]
```

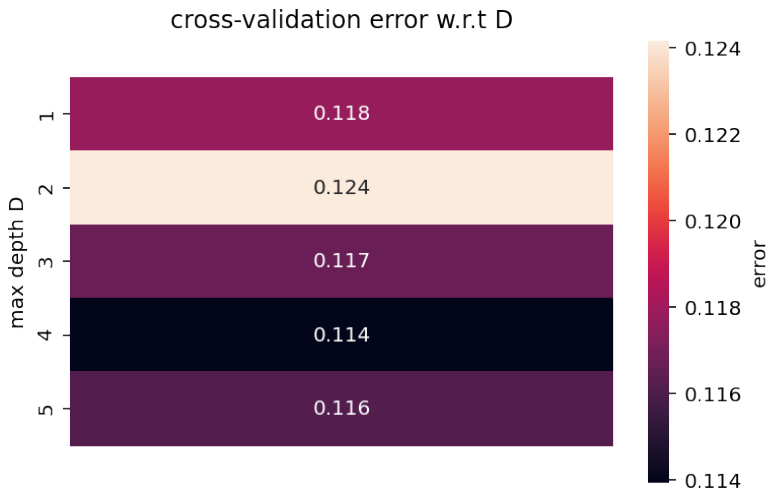
```
Test error =  0.10276243093922655
Test accuracy =  0.8972375690607735
sum of Test accuracy =  [0.887292817679558, 0.8640883977900552, 0.
8972375690607735]

classifier: DecisionTree
BEST_max_depth D is 4
Training error =  0.10197730105434066
Training accuracy =  0.8980226989456593
sum of training accuracy =  [0.9068029425727862, 0.910467461163231
4, 0.8980226989456593]
```



```
Validation error =  0.11393862283456746
Validation accuracy =  0.8860613771654325
sum of Validation accuracy =  [0.8877188357289685, 0.8913172554503
564, 0.8860613771654325]
```



```
Test error =  0.09944751381215466
Test accuracy =  0.9005524861878453
sum of Test accuracy =  [0.887292817679558, 0.8917127071823204, 0.
9005524861878453]
```

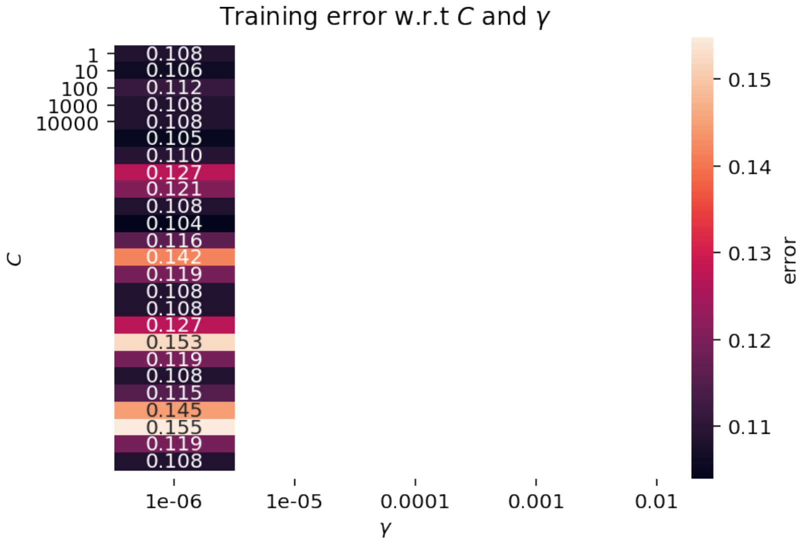


```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.9051206414363727 0.887
260213607615 0.887260213607615
sum_svm= 2.679641068651603
avg_train_knn, avg_test_knn, avg_vali_knn = 0.8962482292594468 0.8
828729281767954 0.8844948533963
sum_knn = 2.663616010832542
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9050977008938923 0.8931
860036832413 0.8883658227815858
sum_dt = 2.6866495273587194
```

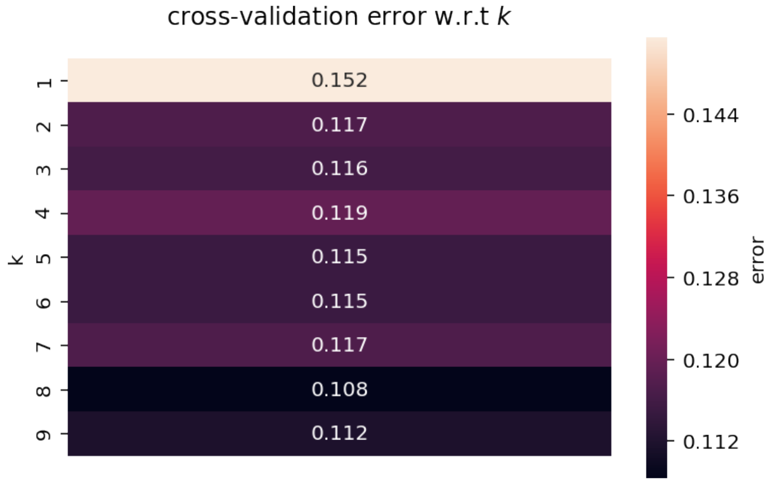
20 train 80 test

In [21]:
```
#20/80
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                    # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]                  # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]              # Last column: Labels.
    Y[Y==0] = -1                    # Convert labels from {0, 1} to {-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.8)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X_test is', X_test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validation error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])
            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])

            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

```
---------- trial: 1 ------------
seed: 18995

The shape of X_train is (904, 16)
The shape of Y_train is (904,)
The shape of X_test is (3617, 16)
The shape of Y_test is (3617,)

classifier: SVM
BEST_C IS 100
BEST_GAMMA IS 1e-06
Training error =  0.09789780151761762
Training accuracy =  0.9021021984823824
sum of training accuracy =  [0.9021021984823824]
Validation error =  0.10397790055248612
Validation accuracy =  0.8960220994475139
sum of Validation accuracy =  [0.8960220994475139]
Test error =  0.1133536079623998
Test error =  0.1133536079623998
Test accuracy =  0.8866463920376002
sum of Test accuracy =  [0.8960220994475139]
```

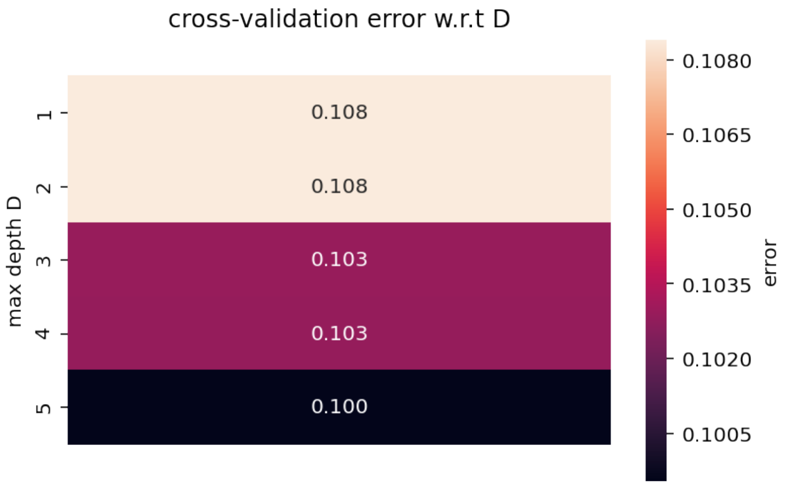


```
classifier: KNN
BEST_K IS 8
Training error =  0.09706639768307324
Training accuracy =  0.9029336023169268
sum of training accuracy =  [0.9029336023169268]
Validation error =  0.10839779005524852
Validation accuracy =  0.8916022099447515
sum of Validation accuracy =  [0.8916022099447515]
Test error =  0.11584185789328172
Test accuracy =  0.8841581421067183
sum of Test accuracy =  [0.8841581421067183]
```

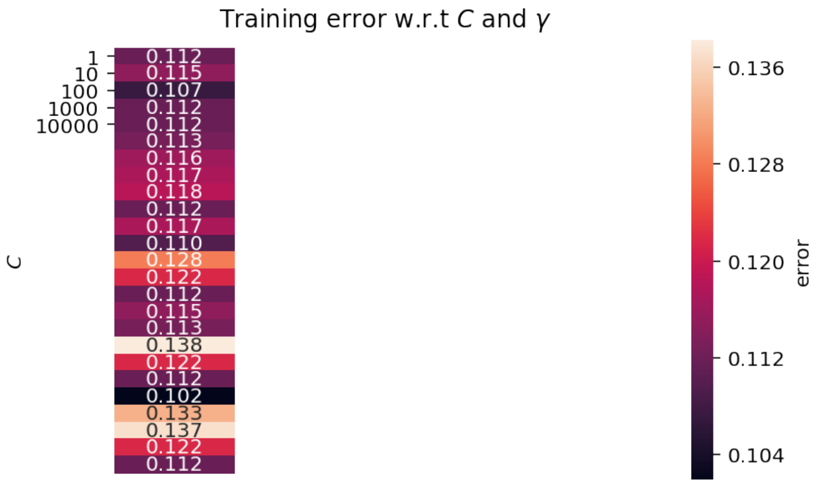


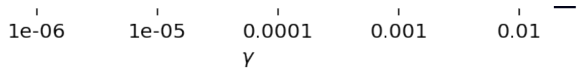
```
classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.07245592719103178
Training accuracy =  0.9275440728089682
sum of training accuracy =  [0.9275440728089682]
```

```
Validation error =  0.09954573357888274
Validation accuracy =  0.9004542664211173
sum of Validation accuracy =  [0.9004542664211173]
Test error =  0.11639480232236665
Test accuracy =  0.8836051976776333
sum of Test accuracy =  [0.8836051976776333]
```

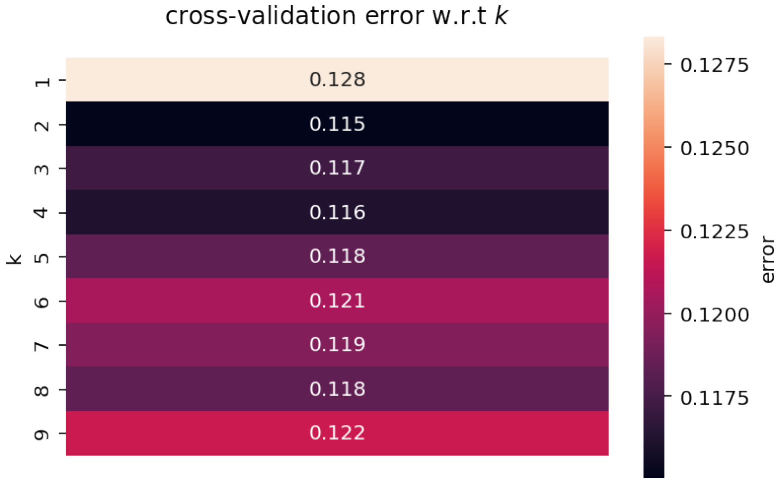


```
---------- trial: 2 ------------
seed: 15736

The shape of X_train is (904, 16)
The shape of Y_train is (904,)
The shape of X_test is (3617, 16)
The shape of Y_test is (3617,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-06
Training error =  0.06498895791782255
Training accuracy =  0.9350110420821774
sum of training accuracy =  [0.9021021984823824, 0.935011042082177
4]
Validation error =  0.10178023327194585
Validation accuracy =  0.8982197667280541
sum of Validation accuracy =  [0.8960220994475139, 0.8982197667280
541]
Test error =  0.11888305225324858
Test accuracy =  0.8811169477467514
sum of Test accuracy =  [0.8960220994475139, 0.8982197667280541]
```

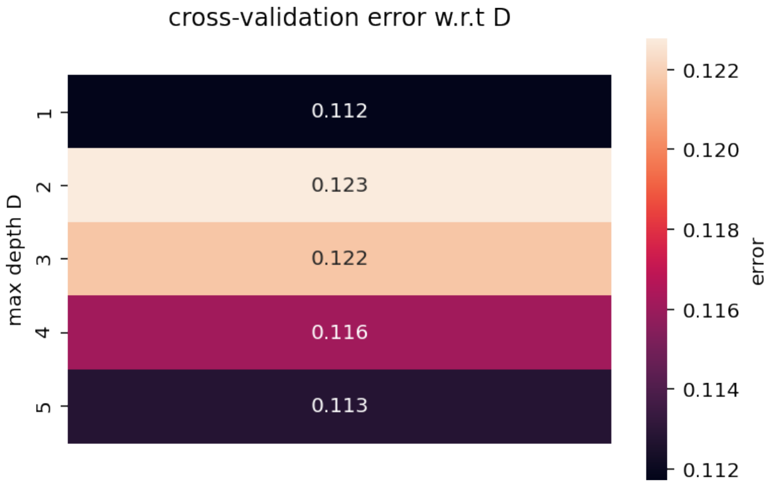




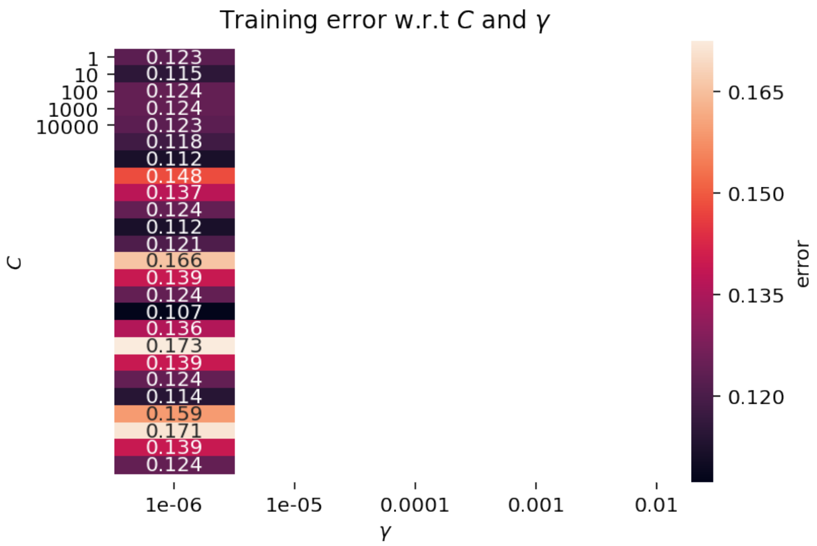
```
classifier: KNN
BEST_K IS 2
Training error =  0.07162681583029595
Training accuracy =  0.928373184169704
sum of training accuracy =  [0.9029336023169268, 0.928373184169704
]
Validation error =  0.11505217925107447
Validation accuracy =  0.8849478207489255
sum of Validation accuracy =  [0.8916022099447515, 0.8849478207489
255]
Test error =  0.11943599668233334
Test accuracy =  0.8805640033176666
sum of Test accuracy =  [0.8841581421067183, 0.8805640033176666]
```



```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.1117256214514416
Training accuracy =  0.8882743785485584
sum of training accuracy =  [0.9275440728089682, 0.888274378548558
4]
Validation error =  0.11172498465316139
Validation accuracy =  0.8882750153468386
sum of Validation accuracy =  [0.9004542664211173, 0.8882750153468
386]
Test error =  0.11611833010782413
Test accuracy =  0.8838816698921759
sum of Test accuracy =  [0.8836051976776333, 0.8838816698921759]
```

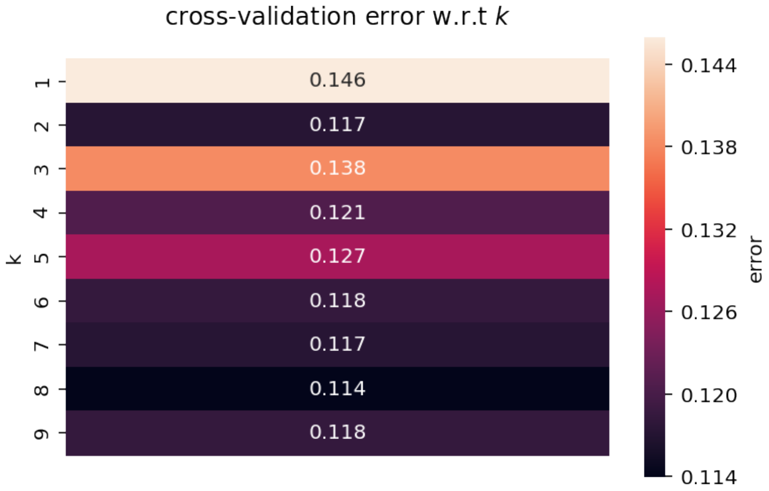


```
---------- trial: 3 ------------
seed: 6044

The shape of X_train is (904, 16)
The shape of Y_train is (904,)
The shape of X_test is (3617, 16)
The shape of Y_test is (3617,)

classifier: SVM
BEST_C IS 1000
BEST_GAMMA IS 1e-06
Training error =  0.09374842392425664
Training accuracy =  0.9062515760757434
sum of training accuracy =  [0.9021021984823824, 0.935011042082177
4, 0.9062515760757434]
Validation error =  0.10728667894413757
Validation accuracy =  0.8927133210558624
sum of Validation accuracy =  [0.8960220994475139, 0.8982197667280
541, 0.8927133210558624]
Test error =  0.11390655239148462
Test accuracy =  0.8860934476085154
sum of Test accuracy =  [0.8960220994475139, 0.8982197667280541, 0
.8927133210558624]
```

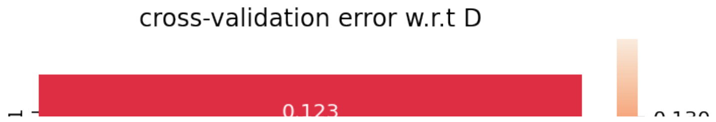


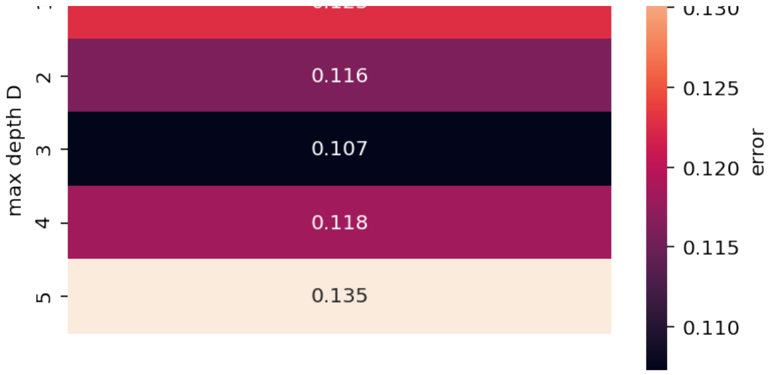
```
classifier: KNN
```

```
classifier: KNN
BEST_K IS 8
Training error =  0.10812911212489396
Training accuracy =  0.891870887875106
sum of training accuracy =  [0.9029336023169268, 0.928373184169704
, 0.891870887875106]
Validation error =  0.11394106813996319
Validation accuracy =  0.8860589318600368
sum of Validation accuracy =  [0.8916022099447515, 0.8849478207489
255, 0.8860589318600368]
Test error =  0.1163948023223665
Test accuracy =  0.8836051976776333
sum of Test accuracy =  [0.8841581421067183, 0.8805640033176666, 0
.8836051976776333]
```



```
classifier: DecisionTree
BEST_max_depth D is 3
Training error =  0.09236606221773924
Training accuracy =  0.9076339377822608
sum of training accuracy =  [0.9275440728089682, 0.888274378548558
4, 0.9076339377822608]
Validation error =  0.10728054020871691
Validation accuracy =  0.8927194597912831
sum of Validation accuracy =  [0.9004542664211173, 0.8882750153468
386, 0.8927194597912831]
Test error =  0.1163948023223665
Test accuracy =  0.8836051976776333
sum of Test accuracy =  [0.8836051976776333, 0.8838816698921759, 0
.8836051976776333]
```





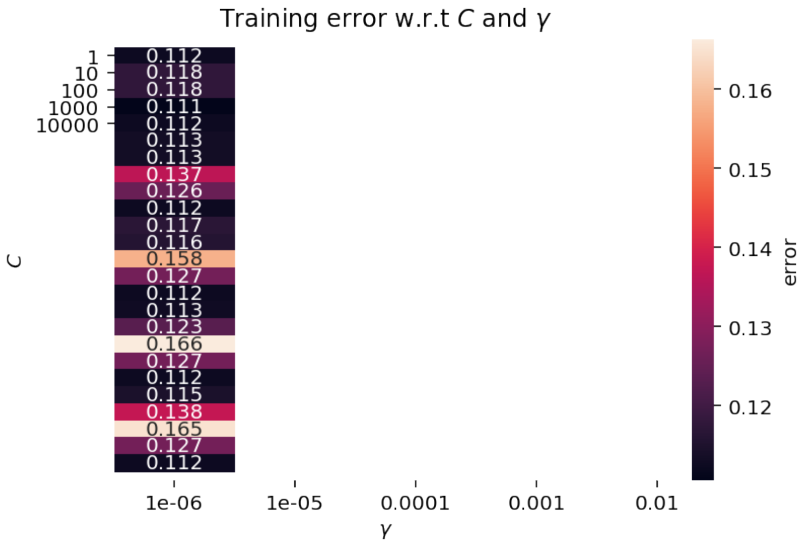
```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.914454938880101 0.8956
517290771435 0.8956517290771435
sum_svm= 2.705758397034388
avg_train_knn, avg_test_knn, avg_vali_knn = 0.9077258914539122 0.8
82775781034006 0.8875363208512379
sum_knn = 2.678037993339156
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9078174630465958 0.8836
973550824808 0.893816247186413
sum_dt = 2.6853310653154896
```

50 train 50 test

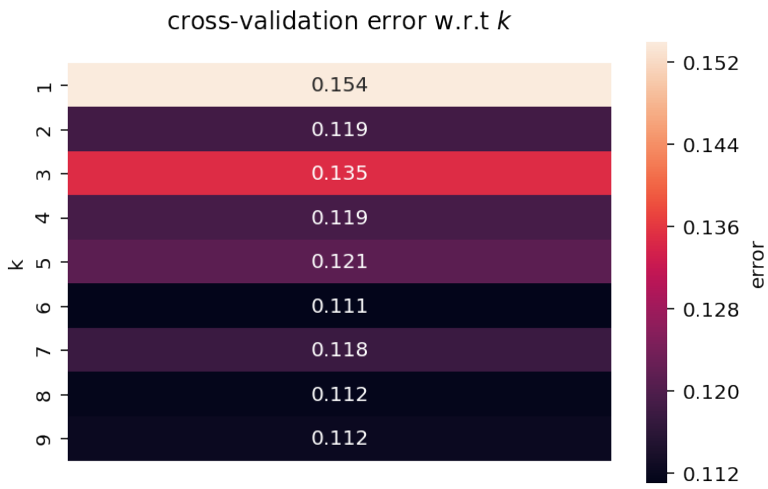
In [22]:

```
#50/50
clf = ['SVM','KNN','DecisionTree']

sum_train_ac_svm = []
sum_test_ac_svm = []
sum_vali_ac_svm = []

sum_train_ac_knn = []
sum_test_ac_knn = []
sum_vali_ac_knn = []

sum_train_ac_dt = []
sum_test_ac_dt = []
sum_vali_ac_dt = []


for i in range(3):
    print('\n---------- trial:', i+1, '------------')
    seed = random.randint(1,50000)
    print('seed:',seed)
    np.random.seed(0)                    # Set the random seed.
    np.random.shuffle(X_and_Y)
    X = X_and_Y[:, 0:-1]                  # First column to second last
column: Features.
    print()
    Y = X_and_Y[:, -1]                   # Last column: Labels.

    Y[Y==0] = -1                         # Convert labels from {0, 1} to
{-1, 1}.
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size
= 0.5)
    print('The shape of X_train is', X_train.shape)
    print('The shape of Y_train is', Y_train.shape)
    print('The shape of X test is', X test.shape)
    print('The shape of Y_test is', Y_test.shape)
    for classifier in clf:
        if classifier == 'SVM':
            print('\nclassifier: SVM')
            opt = SVM(X_train, Y_train, X_test, Y_test)
            print('BEST_C IS', opt[1])
            print('BEST_GAMMA IS', opt[2])

            print('Training error = ', opt[3])
            print('Training accuracy = ', 1- opt[3])
            sum_train_ac_svm.append(1-opt[3])
            print('sum of training accuracy = ',sum_train_ac_svm)

            print('Validation error = ', opt[4])
            print('Validation accuracy = ', 1- opt[4])
            sum_vali_ac_svm.append(1-opt[4])
            print('sum of Validation accuracy = ',sum_vali_ac_svm)

            print("Test error = ", opt[5])
            print("Test accuracy = ", 1- opt[5])
            sum_test_ac_svm.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_svm)

            draw_heatmap_SVM(opt[0], gamma_list, C_list)

        elif classifier == 'KNN':
            print('\nclassifier: KNN')
            opt = KNN(X_train, Y_train, X_test, Y_test)
            print('BEST_K IS', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_knn.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_knn)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_knn.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_knn)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_knn.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_knn)

            draw_heatmap_KNN(opt[0], k_list, title='cross-validation error w.r.t $k$')

        elif classifier == 'DecisionTree':
            print('\nclassifier: DecisionTree')
            opt = DecisionTree(X_train, Y_train, X_test, Y_test)
            print('BEST_max_depth D is', opt[1])

            print('Training error = ', opt[2])
            print('Training accuracy = ', 1- opt[2])
            sum_train_ac_dt.append(1-opt[2])
            print('sum of training accuracy = ',sum_train_ac_dt)

            print('Validation error = ', opt[3])
            print('Validation accuracy = ', 1- opt[3])
            sum_vali_ac_dt.append(1-opt[3])
            print('sum of Validation accuracy = ',sum_vali_ac_dt)

            print("Test error = ", opt[4])
            print("Test accuracy = ", 1- opt[4])
            sum_test_ac_dt.append(1-opt[4])
            print('sum of Test accuracy = ', sum_test_ac_dt)

            draw_heatmap_DT(opt[0], D_list, title='cross-validation
error w.r.t D')

avg_train_svm = np.average(sum_train_ac_svm)
avg_test_svm = np.average(sum_test_ac_svm)
avg_vali_svm = np.average(sum_vali_ac_svm)
print('avg_train_svm,avg_test_svm,avg_vali_svm =',avg_train_svm, av
g_test_svm ,avg_vali_svm)
sum_svm = avg_train_svm + avg_test_svm + avg_vali_svm
print('sum_svm=',sum_svm)

avg_train_knn = np.average(sum_train_ac_knn)
avg_test_knn = np.average(sum_test_ac_knn)
avg_vali_knn = np.average(sum_vali_ac_knn)
print('avg_train_knn, avg_test_knn, avg_vali_knn =', avg_train_knn,
avg_test_knn, avg_vali_knn)
sum_knn = avg_train_knn + avg_test_knn + avg_vali_knn
print('sum_knn =',sum_knn)

avg_train_dt = np.average(sum_train_ac_dt)
avg_test_dt = np.average(sum_test_ac_dt)
avg_vali_dt = np.average(sum_vali_ac_dt)
print('avg_train_dt, avg_test_dt,avg_vali_dt = ',avg_train_dt, avg_
test_dt,avg_vali_dt)
sum_dt = avg_train_dt + avg_test_dt + avg_vali_dt
print('sum_dt =',sum_dt)
```

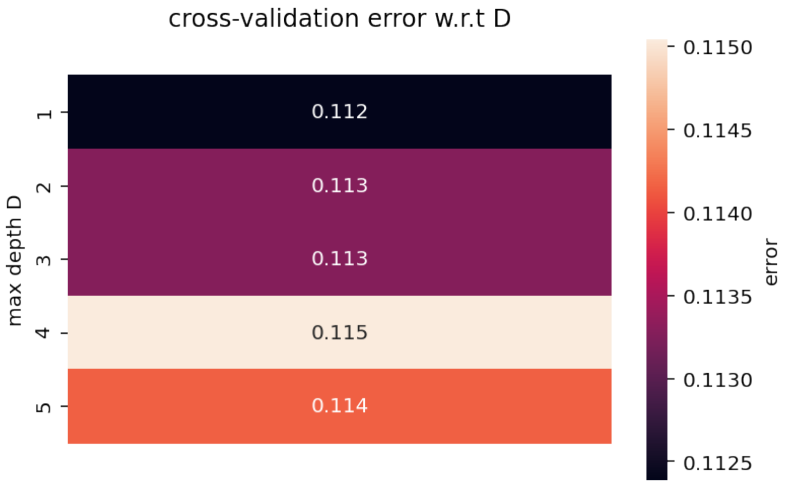
```
---------- trial: 1 ------------
seed: 7527

The shape of X_train is (2260, 16)
The shape of Y_train is (2260,)
The shape of X_test is (2261, 16)
The shape of Y_test is (2261,)

classifier: SVM
BEST_C IS 1
BEST_GAMMA IS 0.001
Training error =  0.03440265486725669
Training accuracy =  0.9655973451327433
Training accuracy =  0.9655973451327433
sum of training accuracy =  [0.9655973451327433]
Validation error =  0.11061946902654862
Validation accuracy =  0.8893805309734514
sum of Validation accuracy =  [0.8893805309734514]
Test error =  0.1194161875276426
Test accuracy =  0.8805838124723574
sum of Test accuracy =  [0.8893805309734514]
```



```
classifier: KNN
BEST_K IS 6
Training error =  0.10011061946902655
Training accuracy =  0.8998893805309734
sum of training accuracy =  [0.8998893805309734]
Validation error =  0.11106194690265492
Validation accuracy =  0.8889380530973451
sum of Validation accuracy =  [0.8889380530973451]
Test error =  0.11764705882352944
Test accuracy =  0.8823529411764706
sum of Test accuracy =  [0.8823529411764706]
```

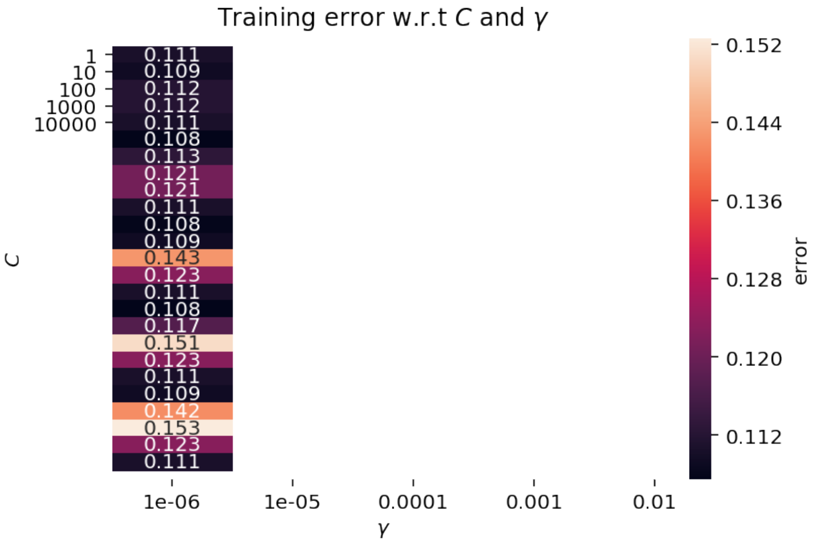


```
classifier: DecisionTree
BEST_max_depth D is 1
Training error =  0.11238938053097347
Training accuracy =  0.8876106194690265
sum of training accuracy =  [0.8876106194690265]
Validation error =  0.11238938053097347
Validation accuracy =  0.8876106194690265
sum of Validation accuracy =  [0.8876106194690265]
Test error =  0.1180893409995577
Test accuracy =  0.8819106590004423
sum of Test accuracy =  [0.8819106590004423]
```

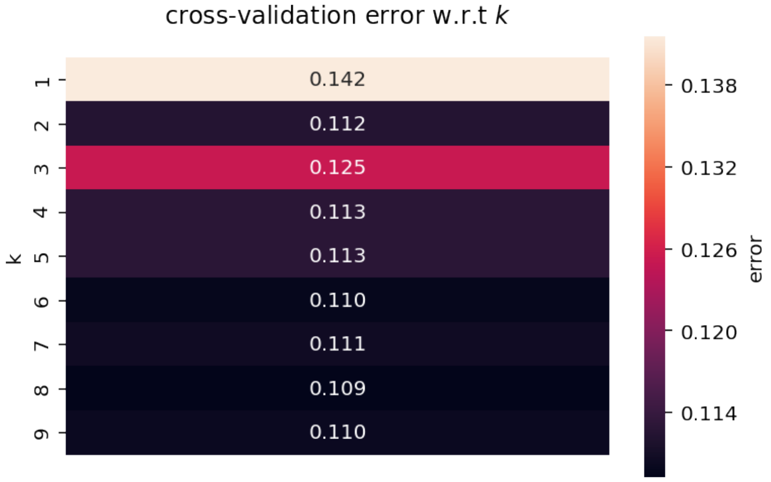


```
---------- trial: 2 ------------
seed: 48321

The shape of X_train is (2260, 16)
The shape of Y_train is (2260,)
The shape of X_test is (2261, 16)
The shape of Y_test is (2261,)

classifier: SVM
BEST_C IS 10
BEST_GAMMA IS 1e-06
Training error =  0.10597345132743352
Training accuracy =  0.8940265486725665
sum of training accuracy =  [0.9655973451327433, 0.894026548672566
5]
Validation error =  0.10752212389380522
Validation accuracy =  0.8924778761061948
sum of Validation accuracy =  [0.8893805309734514, 0.8924778761061
948]
```

```
948]
Test error =  0.12295444493586904
Test accuracy =  0.877045555064131
sum of Test accuracy =  [0.8893805309734514, 0.8924778761061948]
```

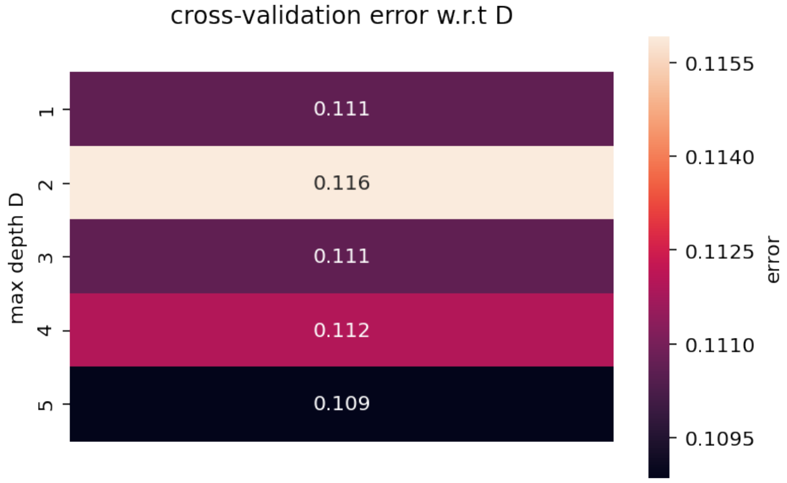


```
classifier: KNN
BEST_K IS 8
Training error =  0.09922566371681418
Training accuracy =  0.9007743362831858
sum of training accuracy =  [0.8998893805309734, 0.900774336283185
8]
Validation error =  0.10929203539823007
Validation accuracy =  0.8907079646017699
sum of Validation accuracy =  [0.8889380530973451, 0.8907079646017
699]
Test error =  0.12383900928792568
Test accuracy =  0.8761609907120743
sum of Test accuracy =  [0.8823529411764706, 0.8761609907120743]
```



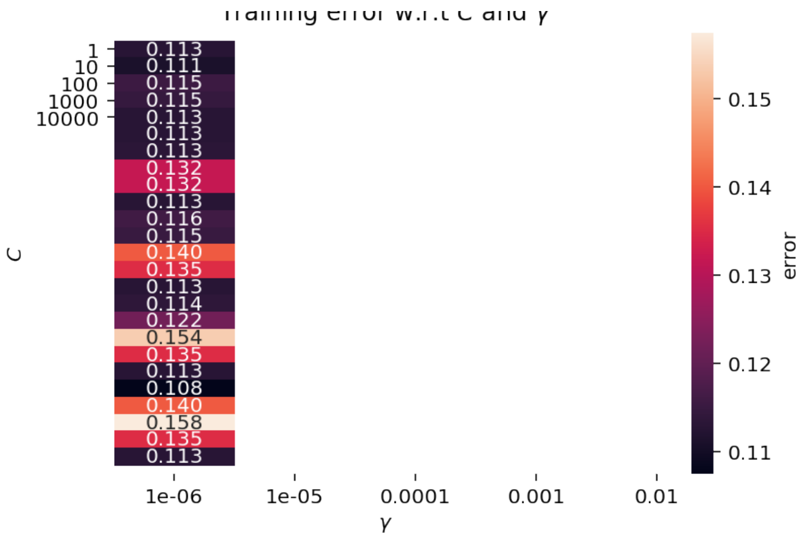
```
classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.0790929203539823
Training accuracy =  0.9209070796460177
sum of training accuracy =  [0.8876106194690265, 0.920907079646017
7]
```

```
Validation error =  0.108849557522124
Validation accuracy =  0.891150442477876
sum of Validation accuracy =  [0.8876106194690265, 0.8911504424778
76]
Test error =  0.11366651923927462
Test accuracy =  0.8863334807607254
sum of Test accuracy =  [0.8819106590004423, 0.8863334807607254]
```

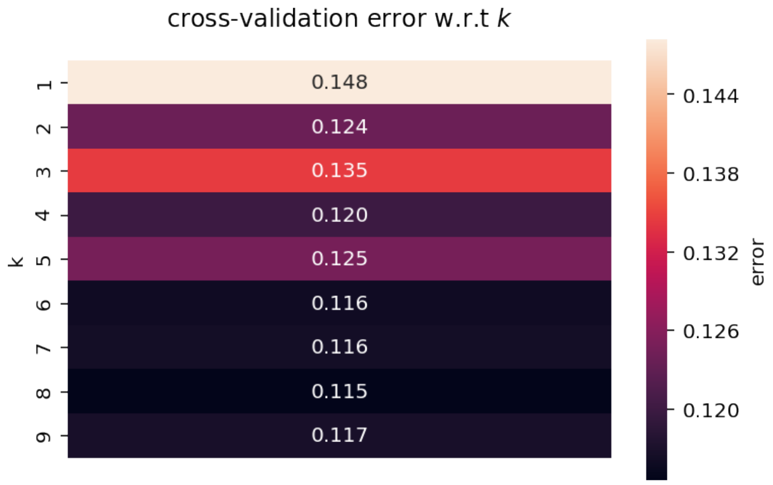

cross-validation error w.r.t D

```
---------- trial: 3 ------------
seed: 46174

The shape of X_train is (2260, 16)
The shape of Y_train is (2260,)
The shape of X_test is (2261, 16)
The shape of Y_test is (2261,)

classifier: SVM
BEST_C IS 10000
BEST_GAMMA IS 1e-06
Training error =  0.09081858407079646
Training accuracy =  0.9091814159292035
sum of training accuracy =  [0.9655973451327433, 0.894026548672566
5, 0.9091814159292035]
Validation error =  0.10752212389380522
Validation accuracy =  0.8924778761061948
sum of Validation accuracy =  [0.8893805309734514, 0.8924778761061
948, 0.8924778761061948]
Test error =  0.11764705882352944
Test accuracy =  0.8823529411764706
sum of Test accuracy =  [0.8893805309734514, 0.8924778761061948, 0
.8924778761061948]
```

Training error w.r.t C and γ

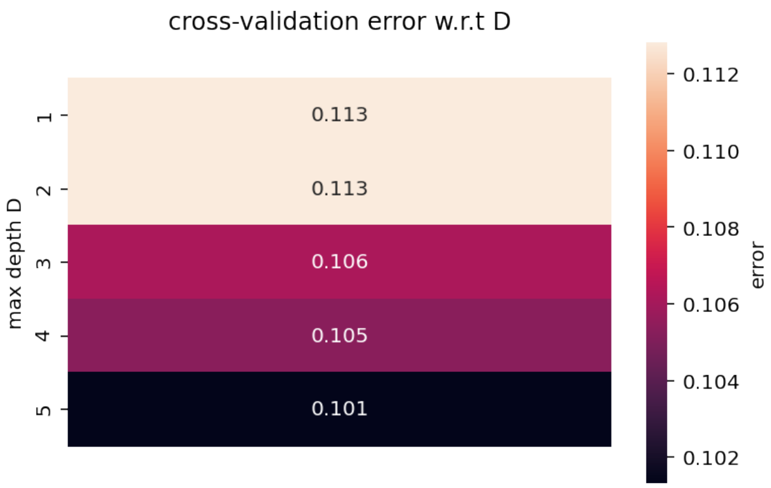


```
classifier: KNN
BEST_K IS 8
Training error =  0.10475663716814165
Training accuracy =  0.8952433628318583
sum of training accuracy =  [0.8998893805309734, 0.900774336283185
8, 0.8952433628318583]
Validation error =  0.1146017699115045
Validation accuracy =  0.8853982300884955
sum of Validation accuracy =  [0.8889380530973451, 0.8907079646017
699, 0.8853982300884955]
Test error =  0.1194161875276426
Test accuracy =  0.8805838124723574
sum of Test accuracy =  [0.8823529411764706, 0.8761609907120743, 0
.8805838124723574]
```



```
classifier: DecisionTree
BEST_max_depth D is 5
Training error =  0.07820796460176971
Training accuracy =  0.9217920353982303
sum of training accuracy =  [0.8876106194690265, 0.920907079646017
7, 0.9217920353982303]
Validation error =  0.10132743362831853
Validation accuracy =  0.8986725663716815
sum of Validation accuracy =  [0.8876106194690265, 0.8911504424778
76, 0.8986725663716815]
Test error =  0.11720477664750106
Test accuracy =  0.8827952233524989
```

```
sum of Test accuracy =  [0.8819106590004423, 0.8863334807607254, 0
.8827952233524989]
```



```
avg_train_svm,avg_test_svm,avg_vali_svm = 0.9229351032448377 0.891
4454277286136 0.8914454277286136
sum_svm= 2.7058259587020648
avg_train_knn, avg_test_knn, avg_vali_knn = 0.8986356932153393 0.8
796992481203008 0.8883480825958703
sum_knn = 2.6666830239315105
avg_train_dt, avg_test_dt,avg_vali_dt =  0.9101032448377581 0.8836
797877045556 0.8924778761061947
sum_dt = 2.6862609086485083
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: