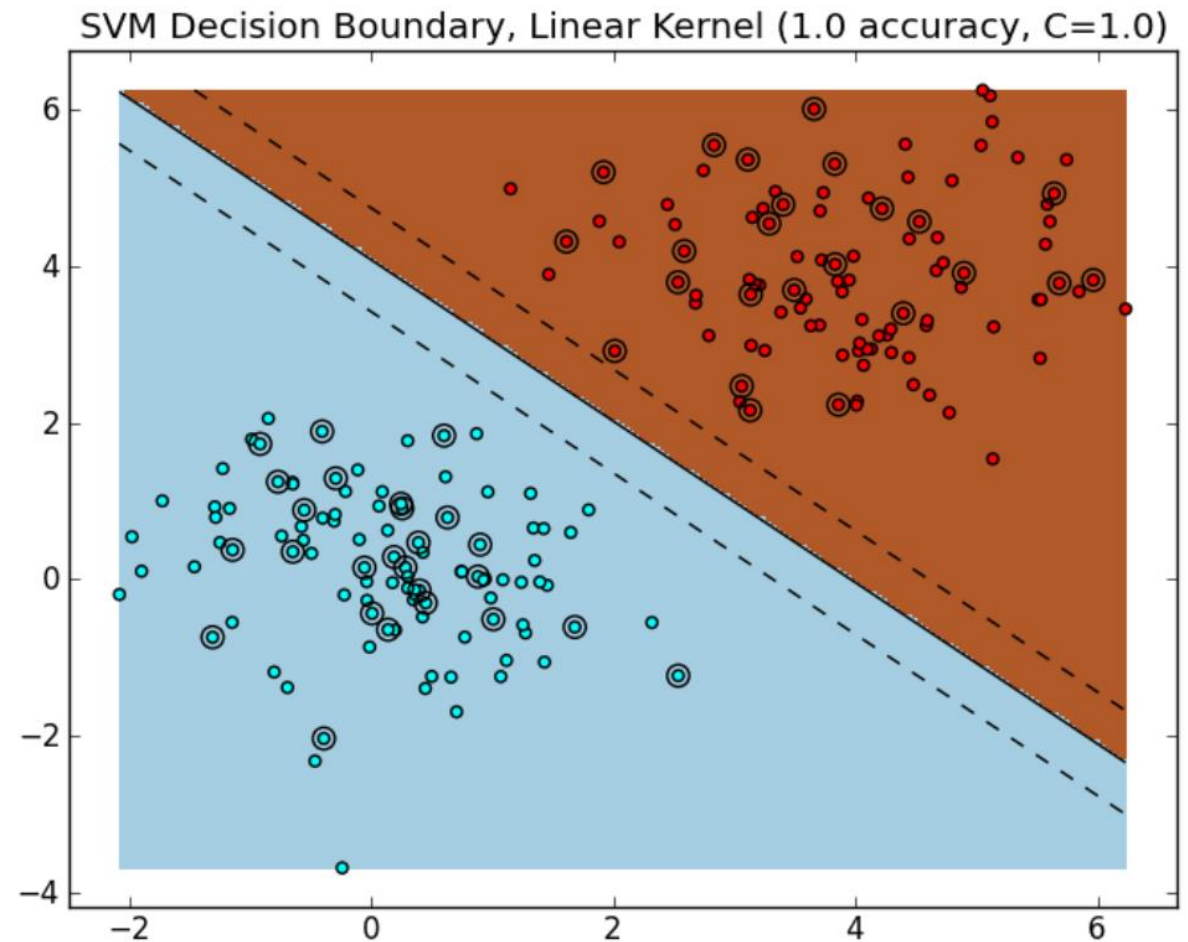
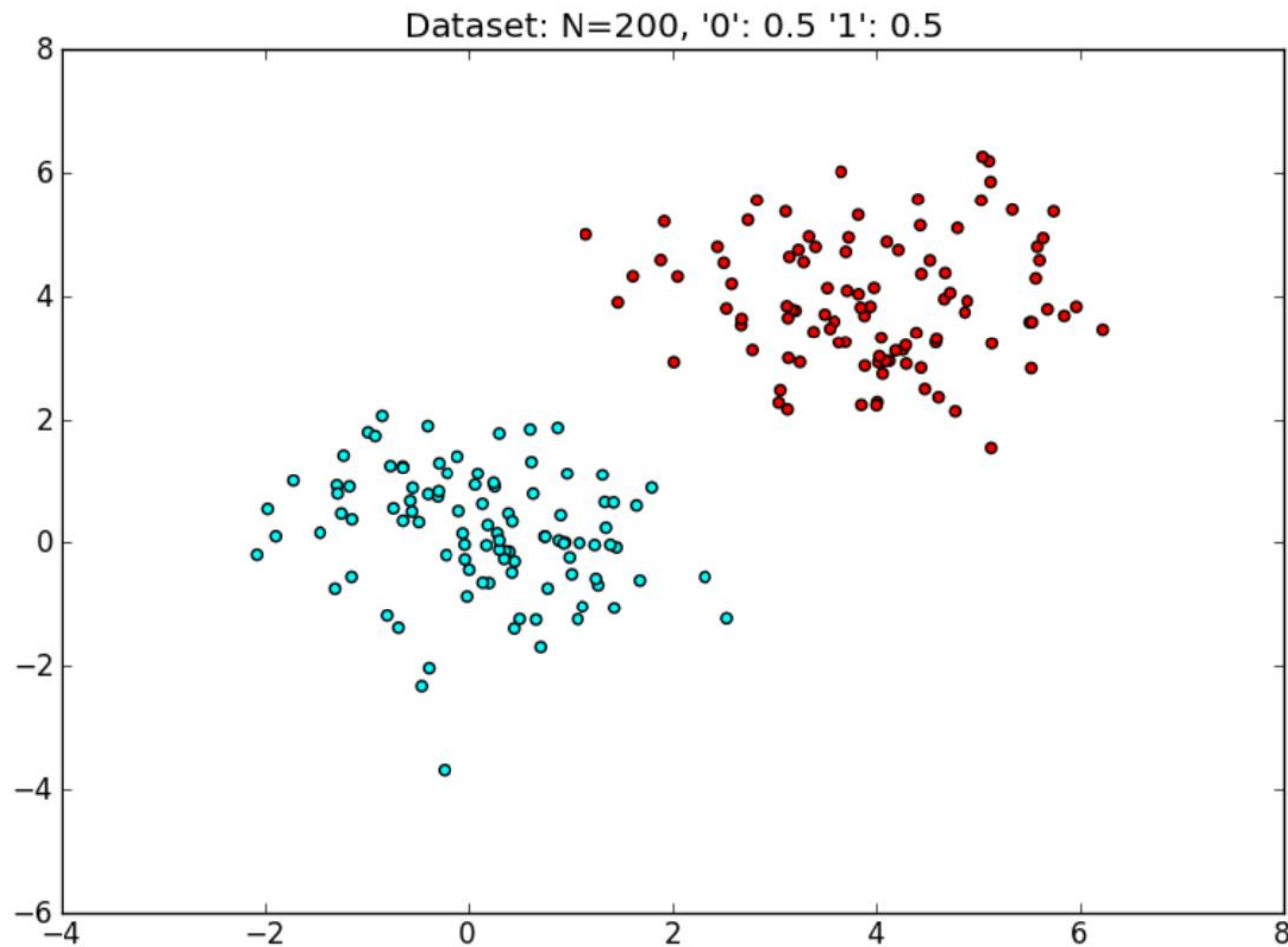


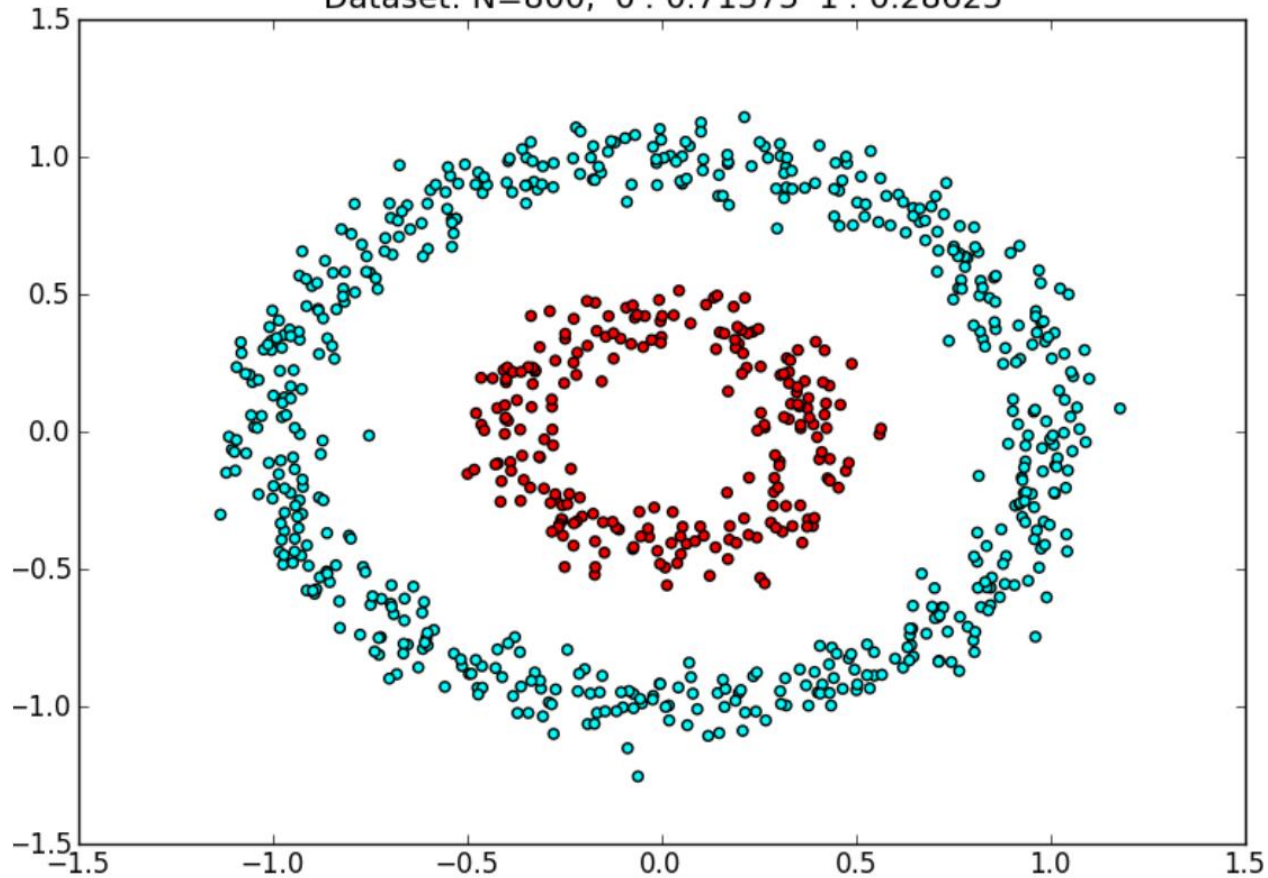
# Support Vector Classifiers

# Linearly separable data

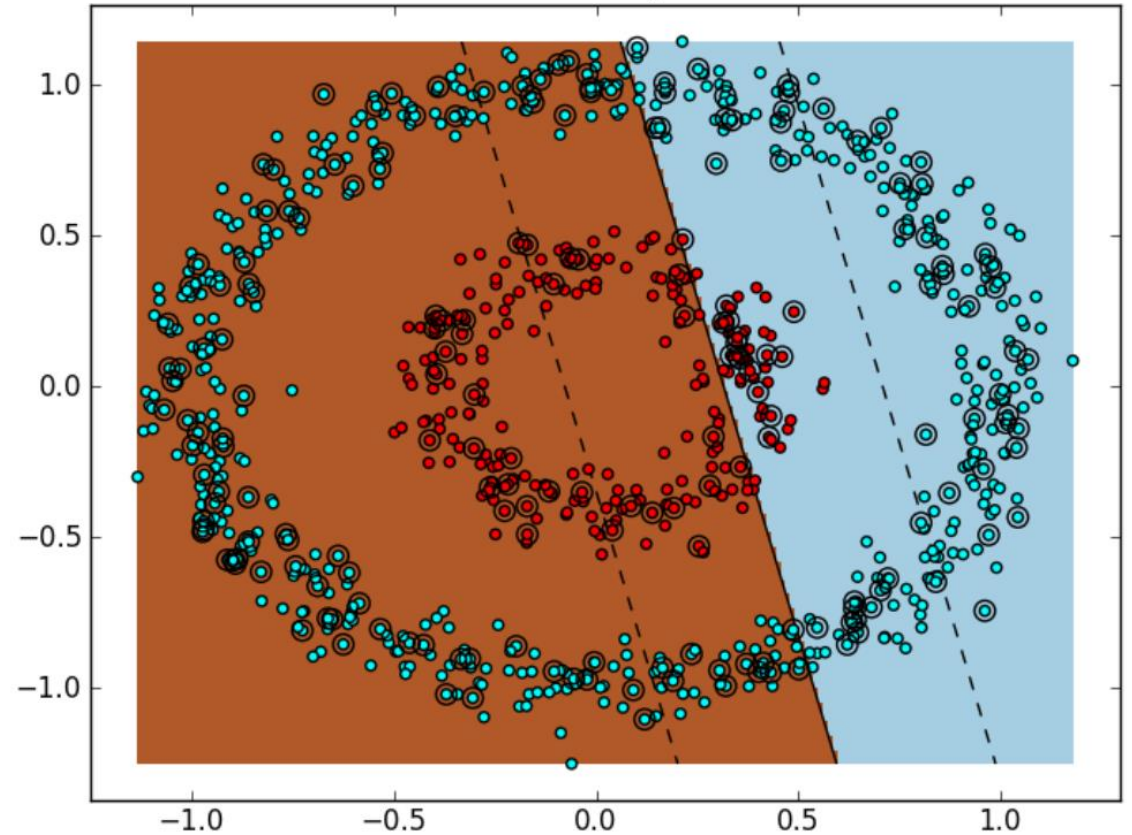


# Linearly nonseparable data

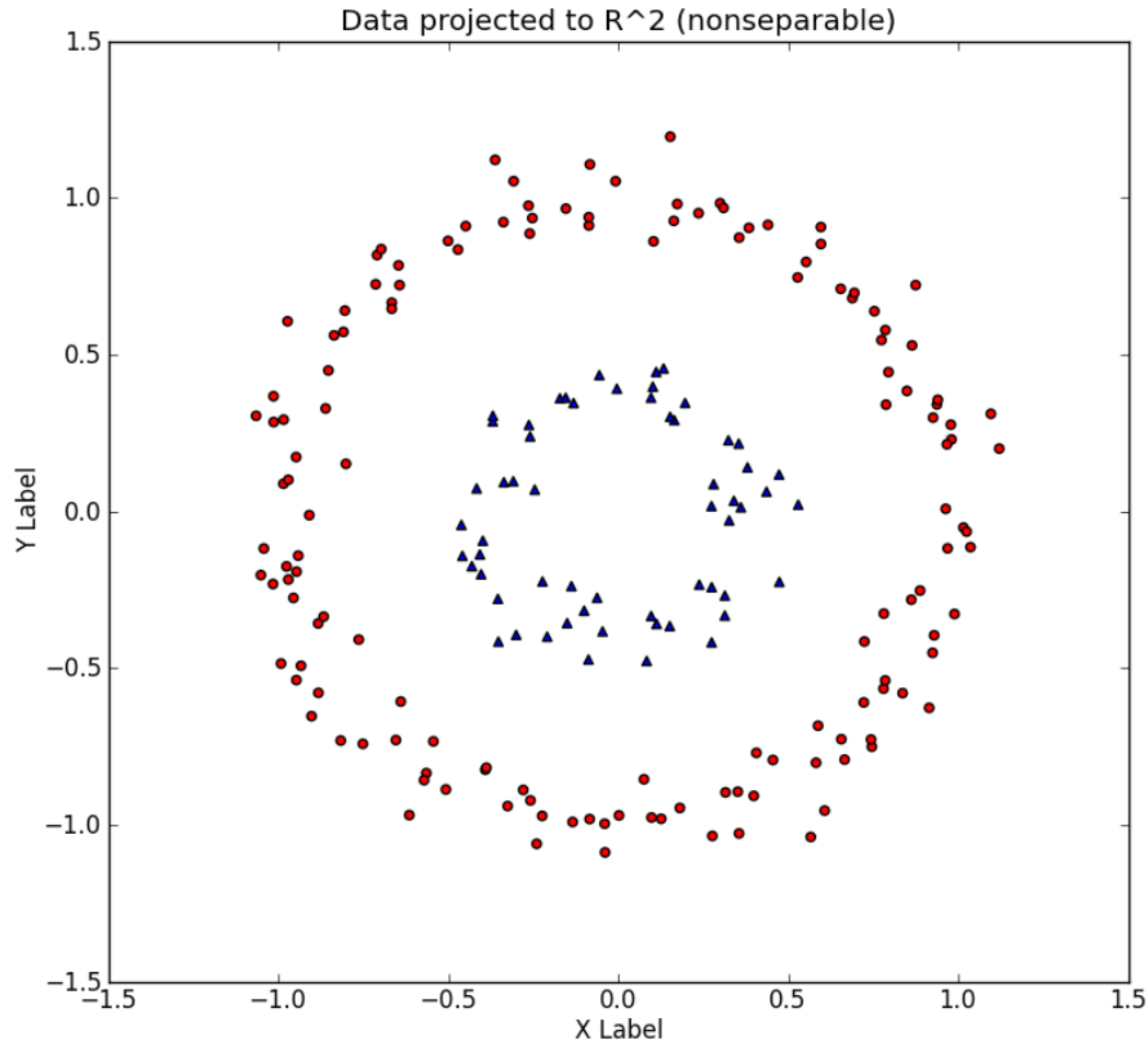
Dataset: N=800, '0': 0.71375 '1': 0.28625



SVM Decision Boundary accuracy=0.445 (Kernel=linear C=1.0)

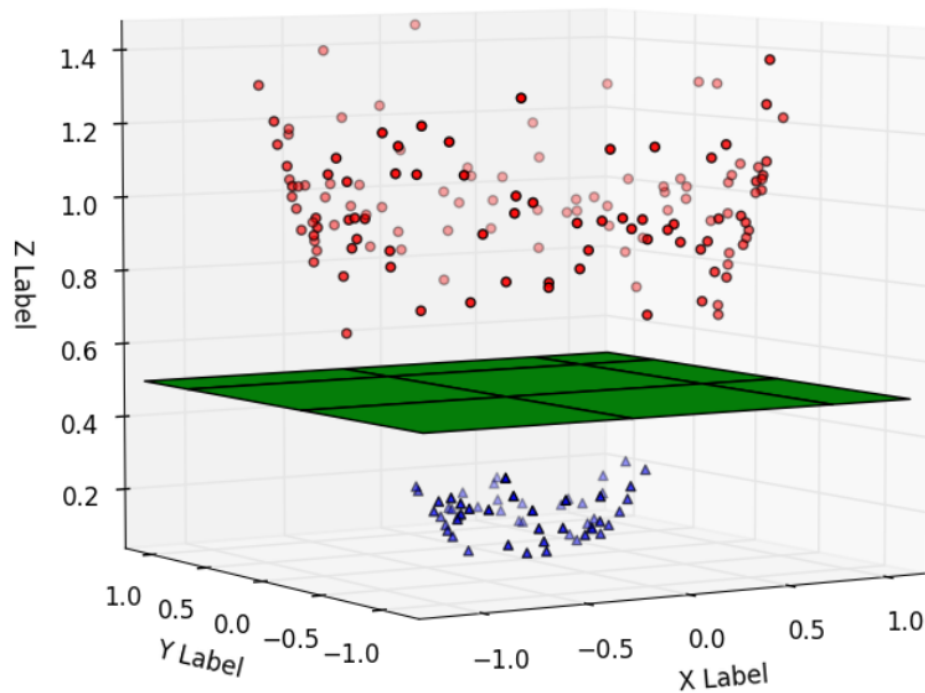


Mathematically transform data (“kernel trick”)  
to be separable in higher dimensions

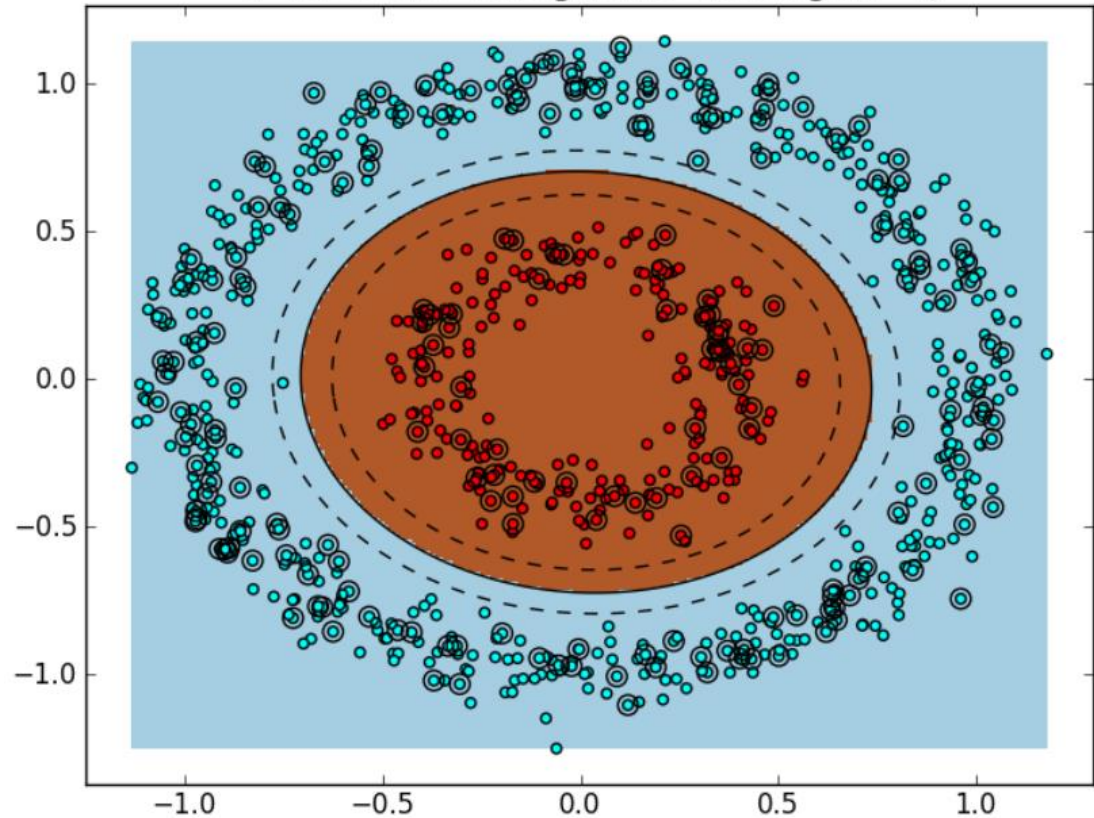


# Hyperplane separates data (looks non-linear in 2-dimensions)

Data in  $R^3$  (separable w/ hyperplane)



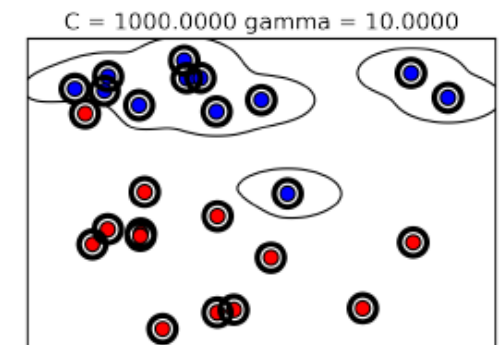
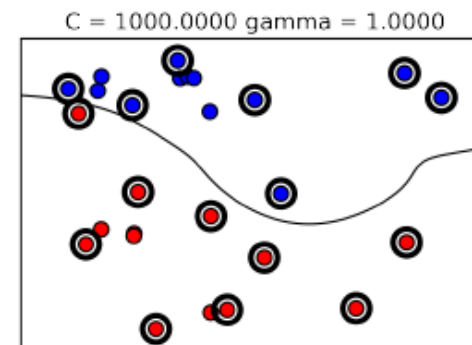
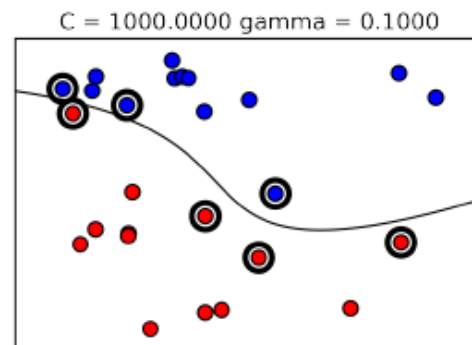
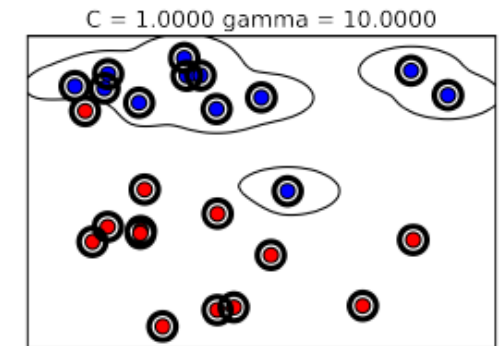
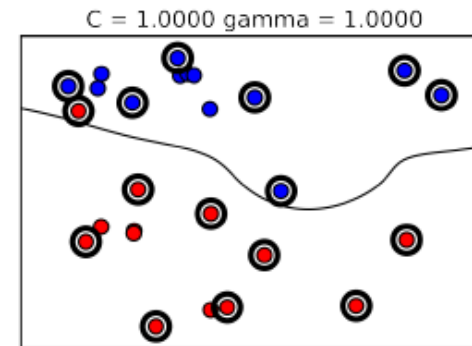
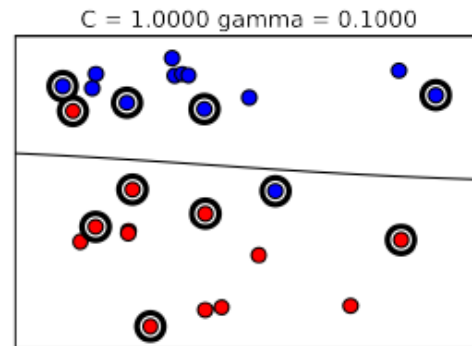
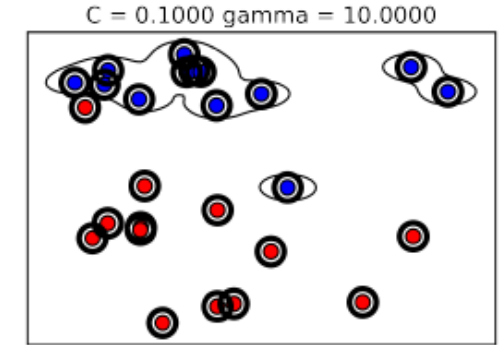
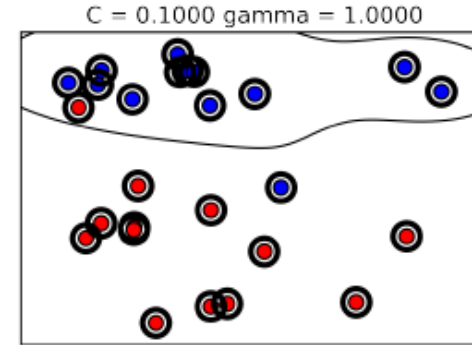
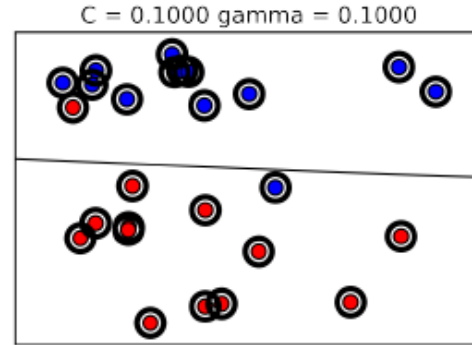
SVM Decision Boundary accuracy=1.0 (Kernel=poly  
C=1.0 coef0=10.0 gamma=0.1 degree=4)





# Tunable Parameters

- $C$  -> controls how strongly each point can distort the decision boundary
- $\gamma$  -> controls radius around each point that excludes the decision boundary



# SVC for Titanic dataset

```
titanic_SVC.py x titanic_SVC_kfold_crossval.py x Amy_PrLD_Linear_SVM_Classifier_WITH_CROSSVALIDATION.py x titanicPerceptron.py x Fishers_exact_GENERIC_DATASET_NO_POOLING.py x compare_PrLD
13 def titanicSVC():
14
15     #combine train and test data into a single dataframe
16     train = pd.read_csv('train.csv')
17     test = pd.read_csv('test.csv')
18     combined = pd.concat([train.drop('Survived', 1), test])
19
20     #drop irrelevant features and features with many NaN values
21     train = train.drop(['PassengerId', 'Name', 'Cabin', 'Ticket', 'Age'], 1)
22
23     #drops two rows with missing values for 'Embarked' feature
24     train = train.dropna(axis=0)
25
26     #pd.get_dummies will convert categorical data into numerical data
27     data = pd.get_dummies(train.drop('Survived', 1))
28
29     #log transform Fare (mean CV score improves ~0.017 and
30     #standard deviation of CV scores improves ~0.02 with log-transformed Fare data)
31     data['Fare'] = np.log10(data['Fare'] + 1)
32
33     #survival is our prediction "target"
34     target = train['Survived']
35
36     #Fit support vector classifier to training data and training targets
37     #C and gamma are adjustable parameters that affect behavior of SVC
38     svc = SVC(C=10, gamma=0.1, random_state=154)
39
40     #Use scikitlearn's cross_val_score method to evaluate the classifier
41     scores = cross_val_score(svc, data, target, cv=10)
```

# titanicSVC results

Accuracy scores from individual models in 10-fold cross validation

True Survivors

119	13
40	51

True Non-survivors

Predicted Survivors    Predicted Non-survivors

```
PS C:\Users\SeanC\Desktop\Python\ML_club> python .\titanic_SVC_kfold_crossval.py

[0.79775281 0.7752809 0.76404494 0.83146067 0.82022472 0.78651685
 0.80898876 0.75280899 0.83146067 0.80681818]

Mean CV score: 0.7975357507660878
Standard dev of CV scores: +/-0.026024573389614107

Confusion matrix:
[[119 13]
 [ 40 51]]
```

