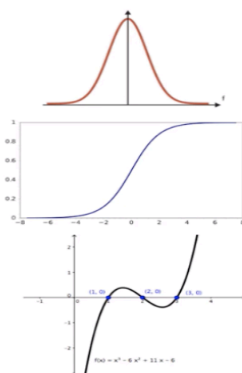


Assignment4: support vector machines (SVM)

In this assignment, you will be using support vector machines (SVM) to build a heart-disease classifier. You need to train/fit SVM classifier using **sklearn** and the heart disease dataset (**Heart.csv**), and then

1. Try different values of C on this dataset and **report the accuracy** with an explanation. For example, you change the value of C in C =1, 50, and 100 and run the SVM training again
2. With an explanation, **report the accuracy** of your model using different kernels such as Radial Basis Function Kernel **rbf** and **linear**.
3. Use 80% of samples as training data size
4. SVM Hyperparameter Tuning using **GridSearchCV**: In order to improve the SVM model accuracy, there are several parameters that need to be tuned. Three major parameters including
 1. **Kernels**: The main function of the kernel is to take low dimensional input space and transform it into a higher-dimensional space. It is mostly useful in non-linear separation problem.



Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}}$$

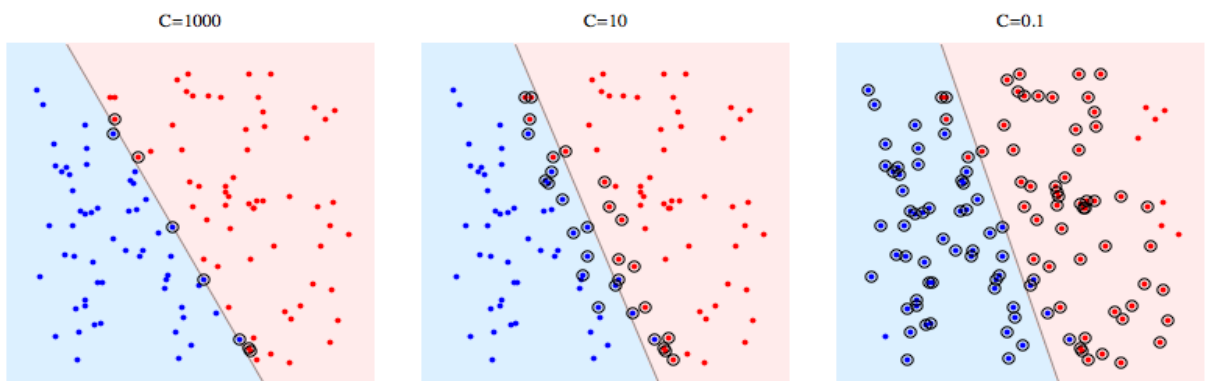
Sigmoid Kernel

$$K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$$

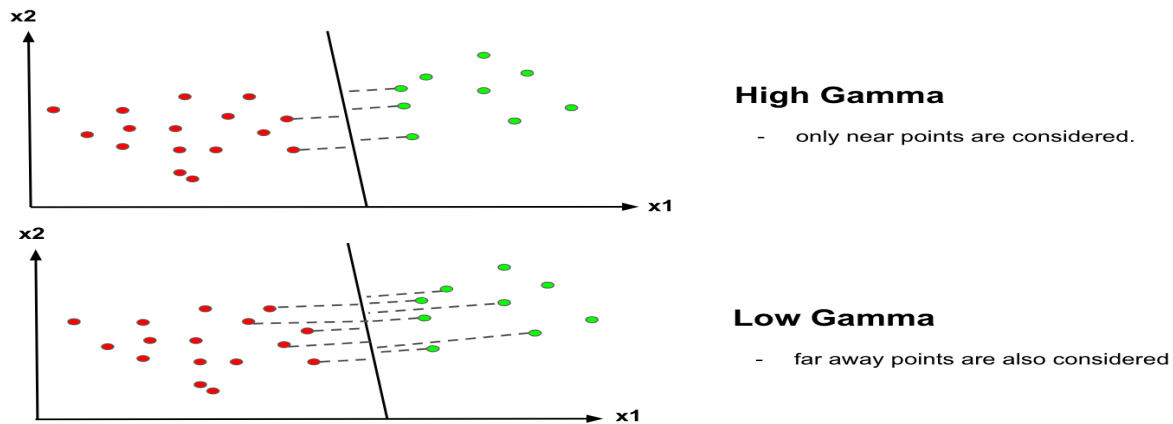
Polynomial Kernel

$$K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

2. **C (Regularization)**: C is the penalty parameter, which represents misclassification or error term. The misclassification or error term tells the SVM optimisation how much error is bearable. when C is high it will classify all the data points correctly, also there is a chance to overfit.



3. **Gamma:** when gamma is higher, nearby points will have high influence; low gamma means far away points also be considered to get the decision boundary.



Tuning the hyper-parameters of an estimator

In scikit-learn, they are passed as arguments to the constructor of the estimator classes. Grid search is commonly used as an approach to hyper-parameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid.

1. Import GridsearchCV from Scikit Learn


```
from sklearn.model_selection import GridSearchCV
```
2. Create a dictionary called param_grid and fill out some parameters for kernels, C and gamma


```
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['rbf', 'poly', 'sigmoid']}
```
3. Create a GridSearchCV object and fit it to the training data


```
grid = GridSearchCV(SVC(), param_grid, refit=True, verbose=2)
grid.fit(X_train, y_train)
```
4. Find the optimal parameters


```
print(grid.best_estimator_)
```
5. Take this grid model to create some predictions using the test set


```
grid_predictions = grid.predict(X_test)
print(confusion_matrix(y_test, grid_predictions))
print(classification_report(y_test, grid_predictions)) #Output
```

What to Submit: Canvas Classroom.

1. Assignment4YourName.pdf (e.g., Assignment-3AliAburas.pdf). That contains all the answers to the above questions + screen-shots of your code! Note: Dont zip the pdf file!
2. Python code