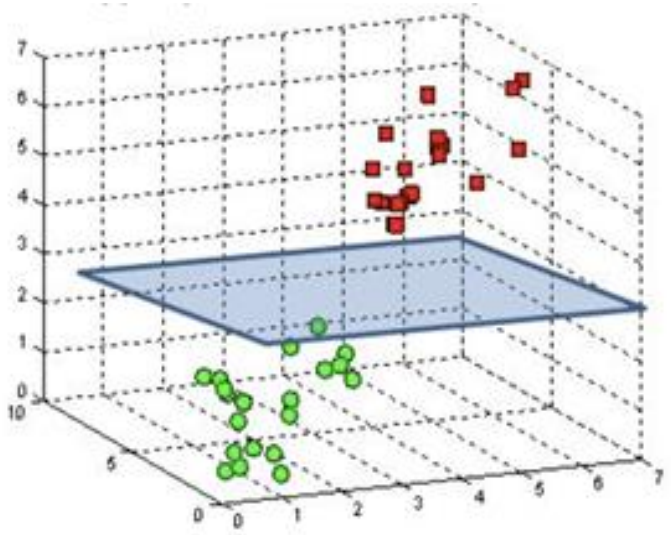
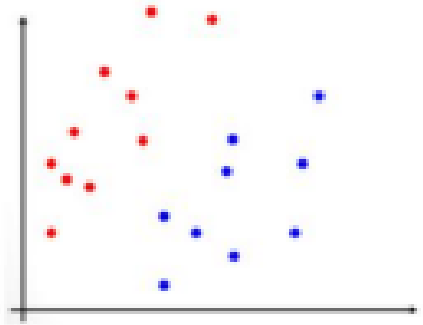
Support Vector Machines
SVMs

Amira Gaber

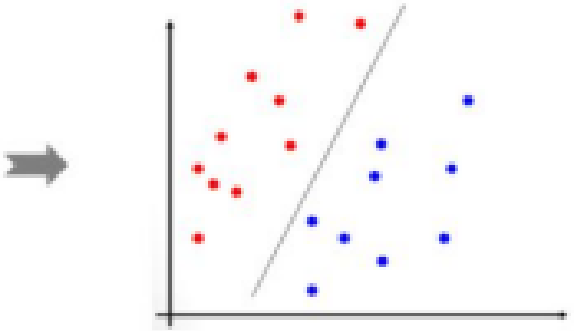
SVM

- It is a supervised machine learning algorithm
- used for both classification and regression problems.
- The main objective of SVM is to separate two classes in the training set with a hyperplane that maximizes the margin between them

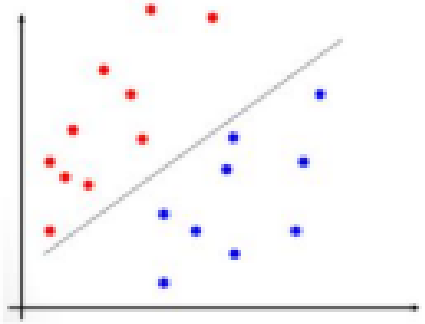


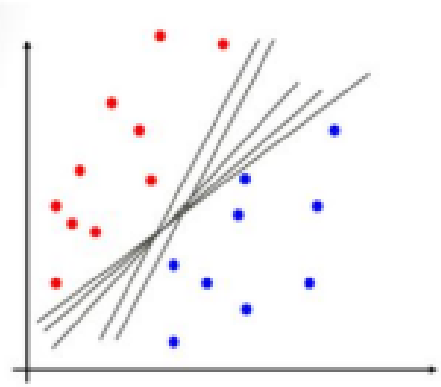


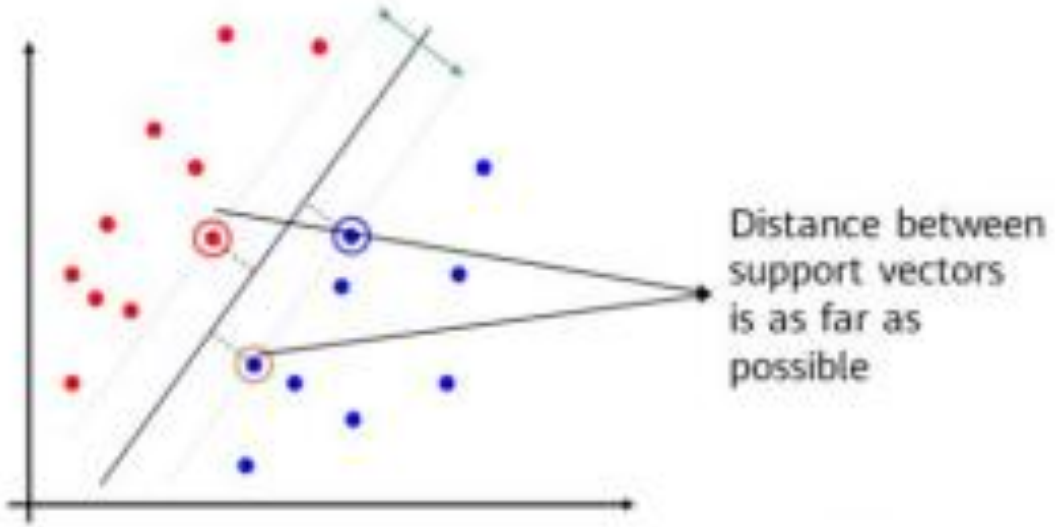
With binary classification
Two-dimensional dataset



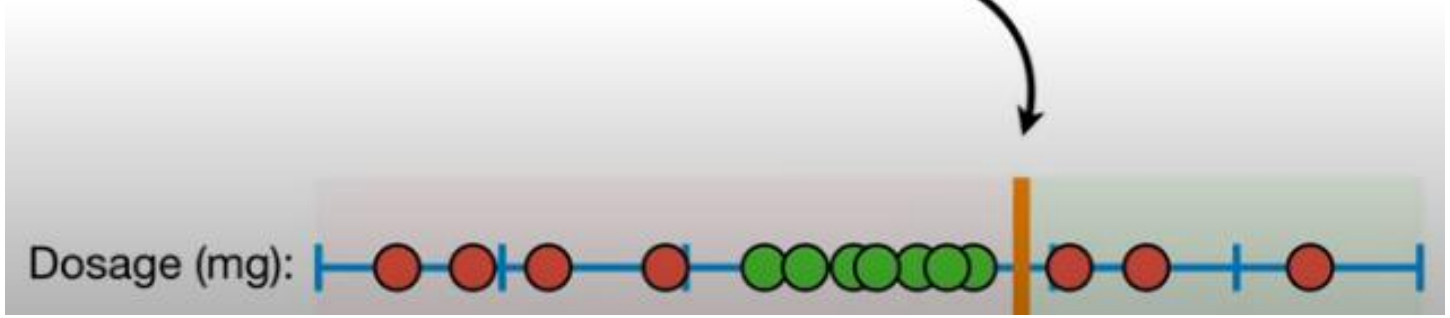
or

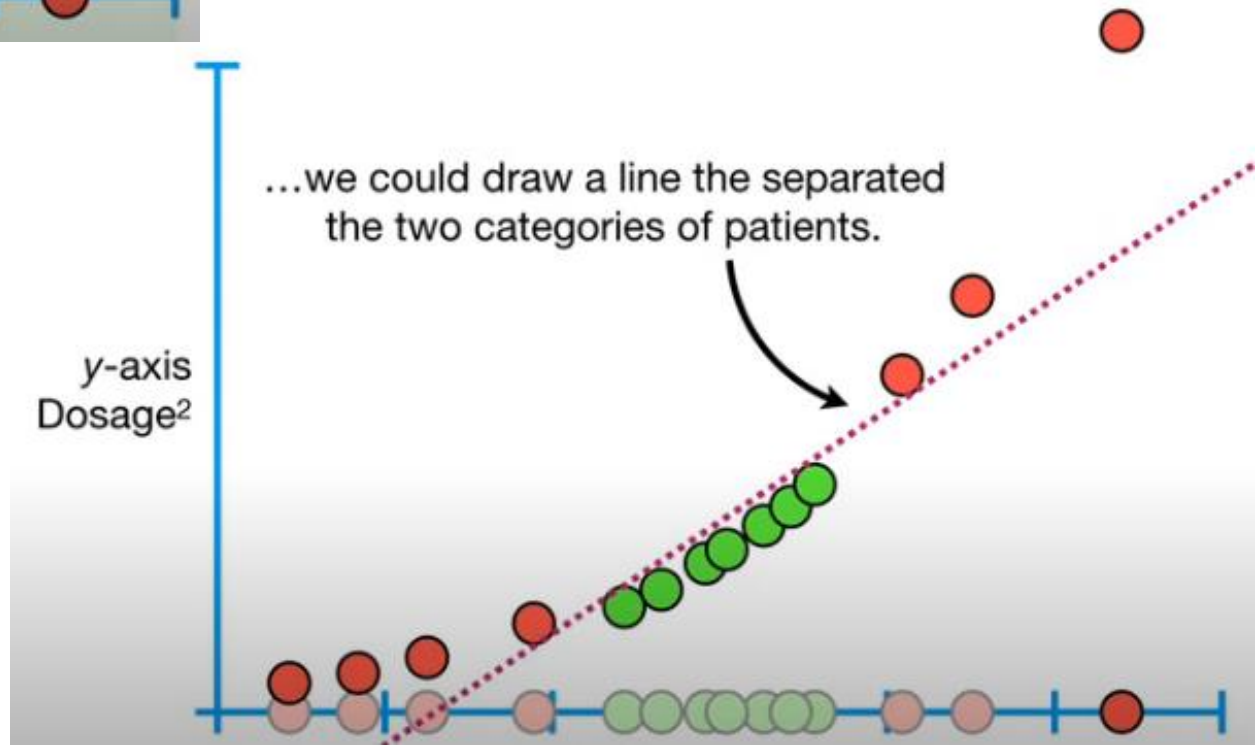






Because this **Training Dataset** had so much overlap, we were unable to find a satisfying **Support Vector Classifier** to separate the patients that were cured from the patients that were not cured.





Kernels

- Kernel models are used on the non-separable dataset to construct non-linear SVMs

- **Linear kernel** is very simple and used for linearly separable data.

- **Polynomial kernel** It has two hyperparameters; the constant term (c), and the degree of the polynomial (d). The linear kernel is a special case from the polynomial kernel with zero constant and degree 1.

- **Radial Basis Function (RBF) kernel** (Gaussian kernel). It is used with a more complicated dataset that could not classified well with the polynomial kernel. There are two parameters for an RBF kernel: C and gamma.

- **Sigmoid kernel**

SVM for multi classification

- The SVM is a binary classification algorithm.

- In multi-classification problems, it is desirable to divide the problem into several binary classification problems.

SVM for multi classification

- **One versus one (OVO):**

For classifying k classes, this method constructs k(k-1)/2 classifiers. Each classifier is trained with just two-class data sets. The unseen sample of test data is fed to the k(k-1)/2 classifiers. If a classifier predict that the sample belongs to certain class, then a vote is added for this class. Finally, the sample was classified with the label with the highest votes from the all classifiers. This method has a limitation when two labels get the same number of votes.

- **One versus all (OVA):**

For classifying k classes, this method constructs k classifiers. Each classifier is trained with the samples between one class and the rest data. The unseen sample of test data is fed to the k classifiers. Then, the sample was classified with the label of the class that has the highest value of the decision.

SVM Advantages

- SVM is one of the best supervised learning algorithms in various application
- SVM has the ability to handle large feature spaces with small number of computations.
- It can deal with a big set of non-linear features.
- It allows to maximizing the generalization ability of a model and prevent the overfitting occurrence.

SVM Disadvantages

- SVM works well with small training dataset but it could not perform well with very large datasets.
- It is very sensitive to noise or any mislabeled data.
- Its performance is very sensitive to the values of its hyperparameters.
- In imbalanced data sets, the SVM does not perform well. It is a challenging problem to correctly classify the class with small number of samples in the training sets.

Implementing SVMs in Python

Steps

1. Import packages and classes

```
from sklearn import datasets          #Import scikit-learn dataset library
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
```

2. Load the dataset

```
cancer = datasets.load_breast_cancer()
```

Implementing SVMs in Python

Steps

3. Explore the dataset (dimensions, describe attributes, plotting data…)

```
# print the names of the 13 features
print("Features: ", cancer.feature_names)
# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)
```

```
Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area' 'mean smoothness' 'mean
compactness' 'mean concavity' 'mean concave points' 'mean symmetry' 'mean fractal dimension'
'radius error' 'texture error' 'perimeter error' 'area error' 'smoothness error' 'compactness error'
'concavity error' 'concave points error' 'symmetry error' 'fractal dimension error' 'worst radius'
'worst texture' 'worst perimeter' 'worst area' 'worst smoothness' 'worst compactness' 'worst
concavity' 'worst concave points' 'worst symmetry' 'worst fractal dimension']

Labels: ['malignant' 'benign']
```

Implementing SVMs in Python

Steps

4. dividing the data into “attributes” and “labels”.

```
cancer.data
cancer.target
```

5. Split the dataset into training and test sets

```
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, test_size=0.3, random_state=109)
# 70% training and 30% test
```

6. Train the SVM model with the training data

```
clf = svm.SVC(kernel='linear') # Linear Kernel
clf.fit(X_train, y_train)
```

Implementing SVMs in Python

Steps

7. Perform prediction on the test set

```
y_pred = clf.predict(X_test)
```

8. Evaluate the performance of the model

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

References

- https://www.youtube.com/playlist?list=PLblh5JKOoLUL3IJ4-yor0HzkqDQ3JmJkc

- https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

Thank You