Image Classification

With Support Vector Machines

Real word uses of image classification.

You interact with image classifiers without even knowing it.

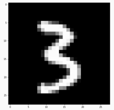
1. Cancer detection, they are used to spot tumors in an MRI or X-Ray.
2. Keeping NSFW content from being uploaded to a social site or forum.
3. Converting handwritten letters to digital text. The postal service uses this every day.

What an image looks like to an ML model.

Normally, images are a two dimensional array of pixels. An X and a Y and a value at that X and Y.

However, ML models cannot accept two dimensional data. So what you must do is flatten the image into a one-dimensional array. A 28x28 == 784x1

28 wide



28 high

784 wide by 1 high

Support Vector Machines

SVM Pros and Cons

PROS:

1. Can be used for Classification or Regression.
2. They are great 'out of the box'. In other words, you usually don't need to tune your hyper-parameters at all.
3. They can handle non-linear data well.
4. They are not prone to outliers.
5. Work well with small data sets.

Great for using when getting correct answer is more important than understanding why you got the correct answer.

CONS:

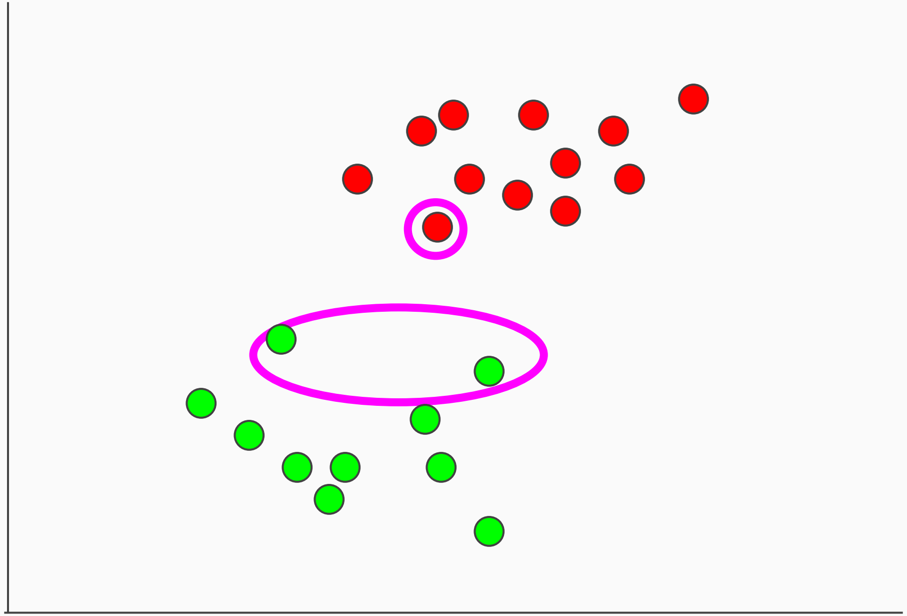
1. They do not give you understanding of your data. SVMs are difficult to interpret and doesn't give you insight as to ***why*** your model works well, IE; they do not give you feature importances.
2. Computationally expensive and slow to train / predict new data.
3. Doesn't scale well to very large data sets. Works better with smaller data due to computation costs.
4. Doesn't give you predicted probability (can only estimate it using Cross Validation techniques).

Support Vectors

Support Vectors are the data points that are the closest to each other of different classes.

Here circled in pink.

Remember, a vector is just a fancy word for a data point.

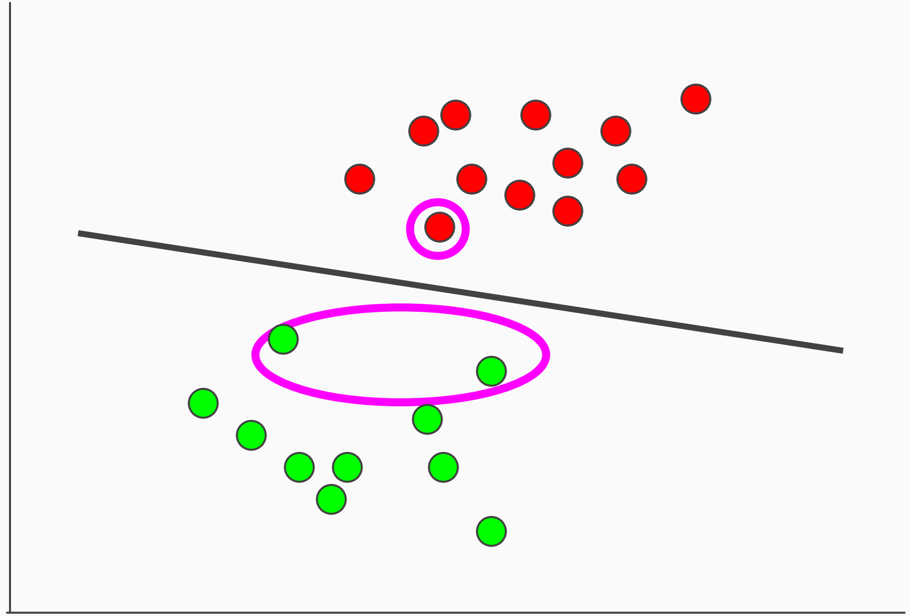


Support Vectors

Support Vectors are the data points that are closest a decision boundary.

It uses these ‘vectors’ to draw a decision boundary.

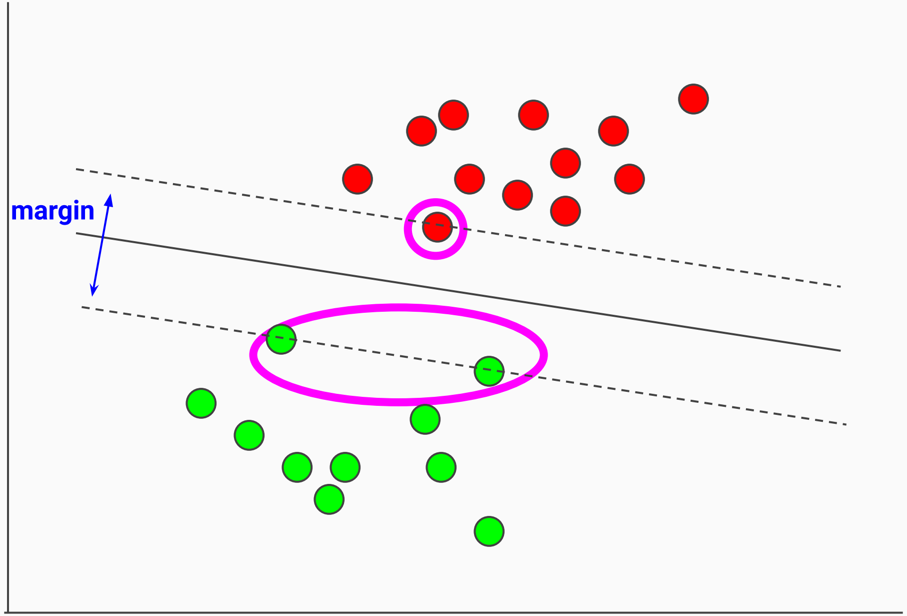
Here in black.



Support Vectors

The margins (here in dotted black) are the distance of the data points the boundary line.

SVMs draw the decision line by finding the maximum margin distance.

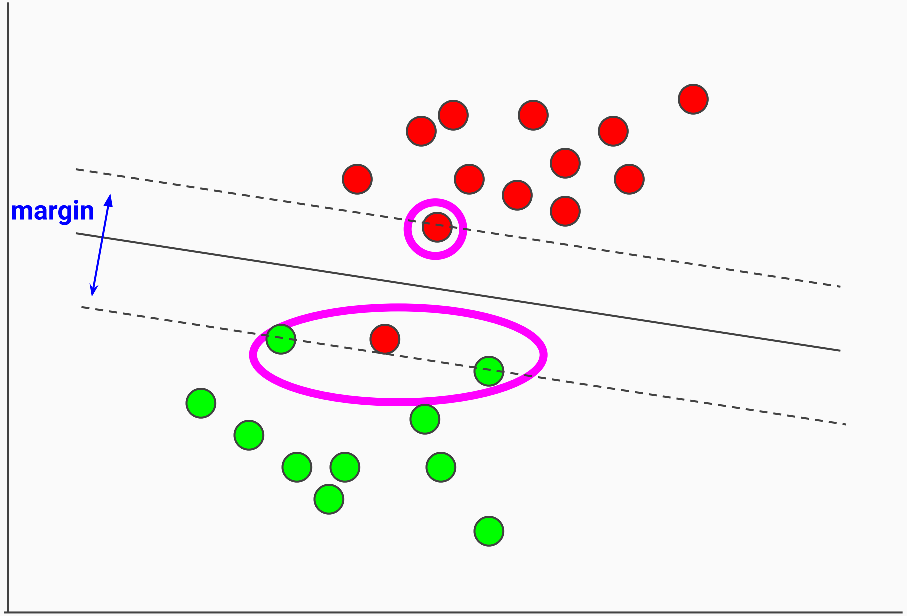


Soft Margins

This being said, they do allow for soft margins, and allow for purposeful misclassifications to handle outliers.

You can set this parameter by setting your C value when initializing your SVM classifier.
C is a misclassification penalty weight.

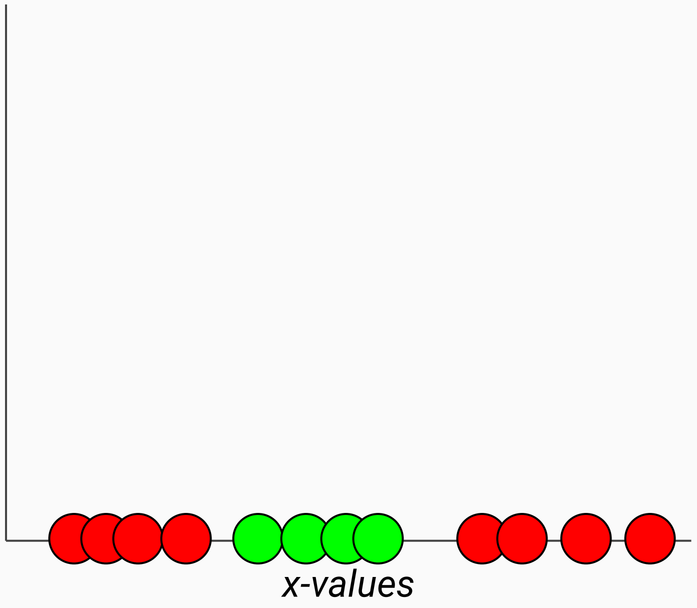
```
from sklearn.svm import SVC
model = SVC(C=2.5)
```



SVMs are ‘tricky’

Kernel Tricks...
Adding dimensionality to your data.

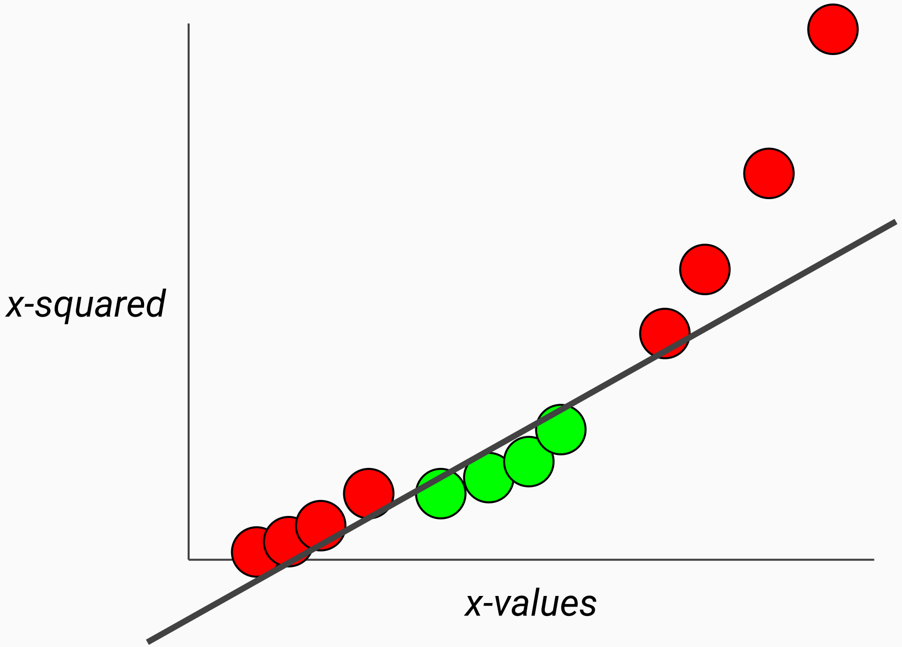
Say we have data that is not linearly separable.



Kernel Tricks...
Adding dimensionality to your data.

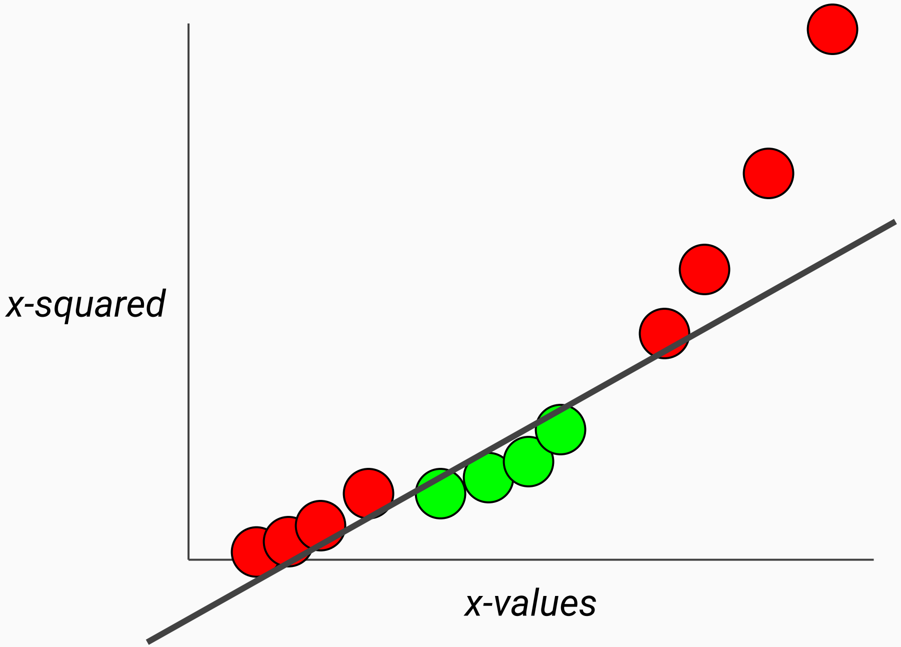
By adding dimensionality to your data, it can figure out new ways to split the data.

Here by taking the square of the x-value, it can now linearly separate the two classes.



Kernel Tricks...
Adding dimensionality to your data.

SVM’s work by moving data into a relatively high dimensional space.



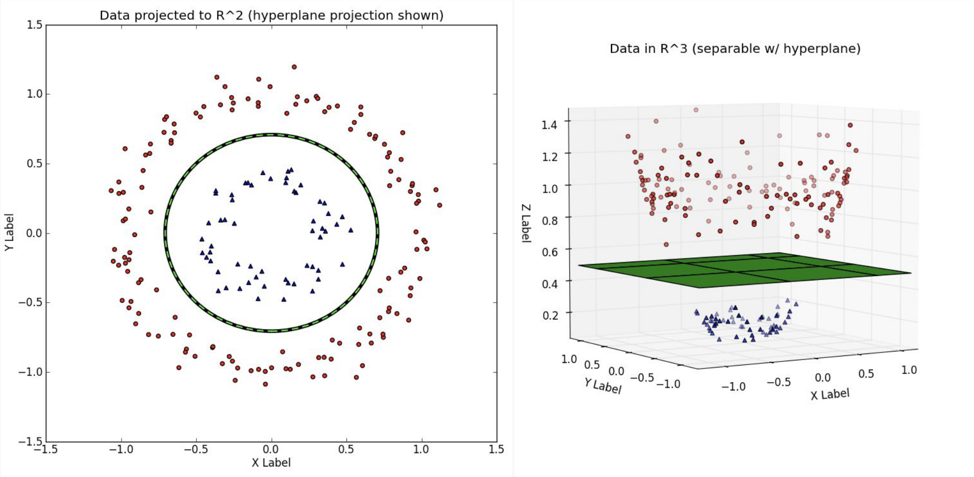
EVEN MORE DIMENSIONS!

SVM’s can add even more dimensions to your data to help with the separation process.

Here, the data is transformed into a 3 dimensions to separate the data.

You can set this parameter in your SVM by.

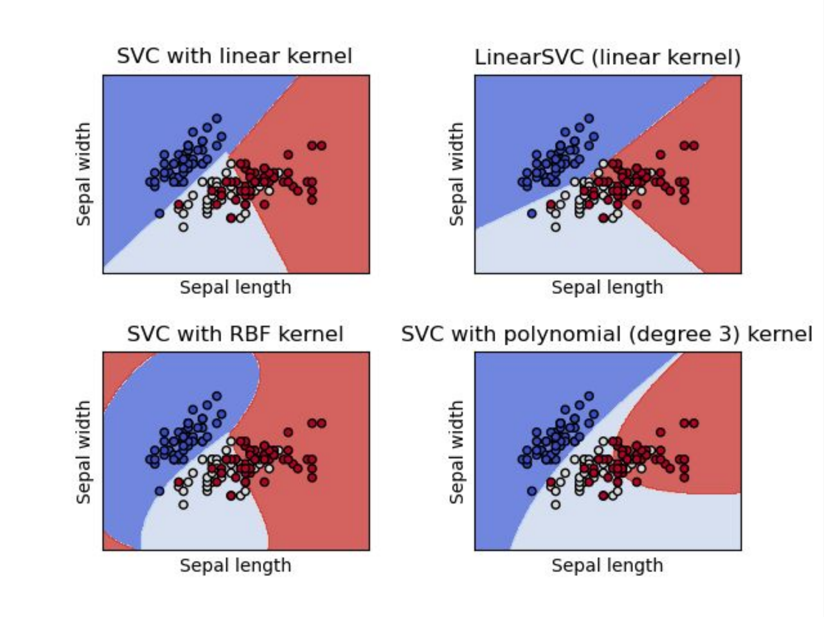
```
from sklearn.svm import SVC
model = SVC(kernel='poly', degree=3)
```



Different ways to add dimension to your data. Are called 'kernel tricks'

The most common kernel tricks in SVMs are.

1. Linear Kernels
2. Polynomial Kernels (link)
3. (RBF) Radial Bias Function Kernels (link)



SVM Pros and Cons

PROS:

1. Can be used for Classification or Regression.
2. They are great 'out of the box'. In other words, you usually don't need to tune your hyper-parameters at all.
3. They can handle non-linear data well.
4. They are not prone to outliers.
5. Work well with small data sets.

Great for using when getting correct answer is more important than understanding why you got the correct answer.

CONS:

1. They do not give you understanding of your data. SVMs are difficult to interpret and doesn't give you insight as to ***why*** your model works well, IE; they do not give you feature importances.
2. Computationally expensive and slow to train / predict new data.
3. Doesn't scale well to very large data sets. Works better with smaller data due to computation costs.
4. Doesn't give you predicted probability (can only estimate it using Cross Validation techniques).

Projects Update

Projects Update

1. MVP / Presentation Due
   a. 12/02 For Wed class
   b. 12/04 For Friday class
2. FINAL FINAL DUE: Demo Night is **December 10th, 2020 from 5:30-7:30.**

Demo Night Lowdown

December 10th, 2020 from 5:30-7:30

Think of demo night as an science fair style event.

You will have **approx 5 mins** to show off your project and answer questions.

As of now, you will be placed into breakout rooms with a few other teams and be showing your projects off there. The event attendees will be moving from room to room.

Demo Night Lowdown

Last years attendees

Amazon, American Express, Beeswax, Bloomberg L.P., CIgna, DCAS, Google, GrantAnswers, Healthify, IBM, J.P. Morgan Chase, FactSet, Outcome Health, Blue State, SevenRooms, NuArch, NYC Mayor's Office, Civic Hall, WW, HBO, CSforALL, Squarespace

Next week:
Project Pitches

An in class 2-3 min project pitch.

These are also listed in todays README.md file.

1. The story/motivation behind why you are doing this particular project.

2. A description of your data.

3. For modeling projects: Your input features and your output prediction

4. For visualization project: Five rough sketches of what visualizations you are going to do and why you are choosing to do these visualizations.

5. What is your final product going to look like.