ML Series

Part 1

Sign-in Sheet with links to ML worksheet
bit.ly/bumicspring2019ws02

MACHINE
INTELLIGENCE
COMMUNITY

2/13/2019

ML Series

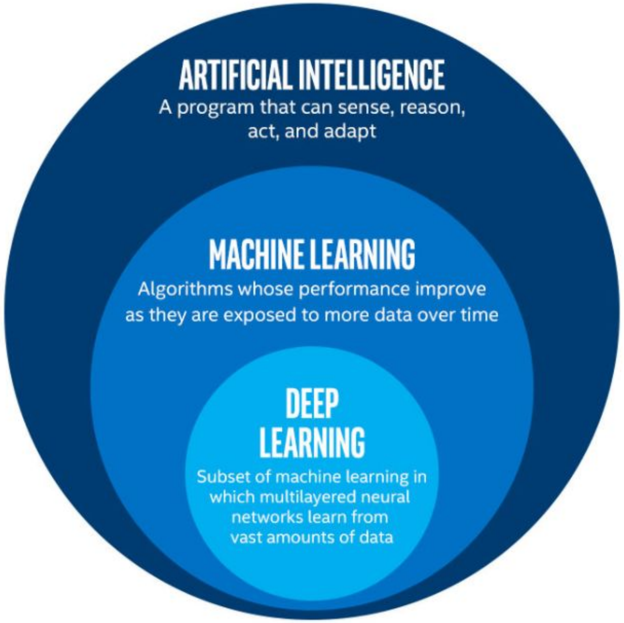
Part 1

Darcy, Duy, Zack

MACHINE INTELLIGENCE COMMUNITY

Traditional machine learning

- Traditional machine learning offer an alternative to the data-and computation-hungry deep learning.
- Traditional machine learning can offer a more interpretable model.



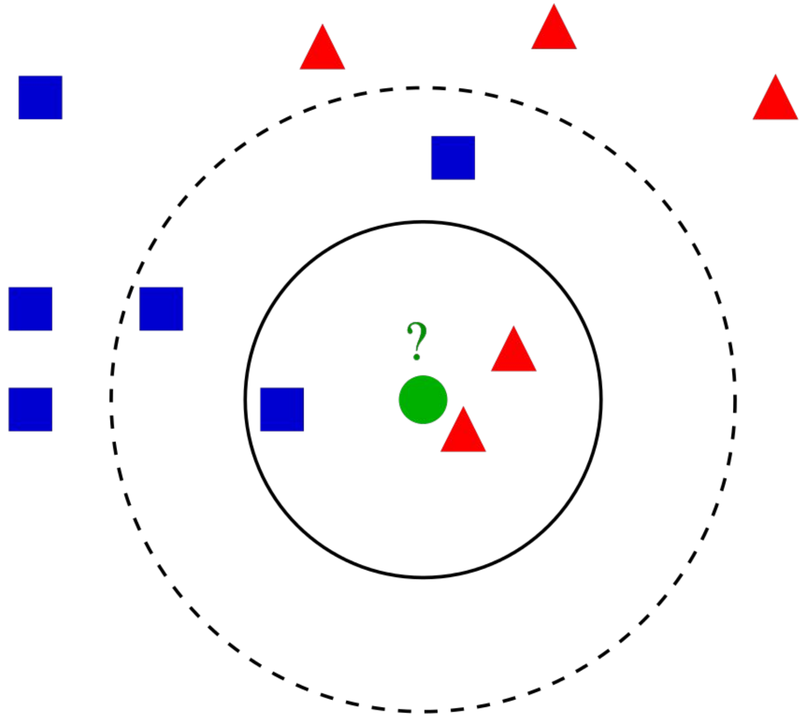
k Nearest Neighbor (kNN)

What’s kNN

- Classify based on the k-nearest neighbor to the value in question.
- K-nearest neighbor doesn’t really need “training”, the classification is left until the value in question need to be classify.
- kNN is parametrized by k (the number of neighbor we need to consider for each value).
    - The smaller k is, the more prone to overfitting.
    - The larger k is, the more smoothing.

kNN in action

- k number of neighboring data point around is selected.
- These k-nearest neighbors vote on the class of the data point being classify.



Distance metrics

- Minkowski (L^p) $D(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$

- Manhattan (L1) $D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$

- Euclidian (L2) $D(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

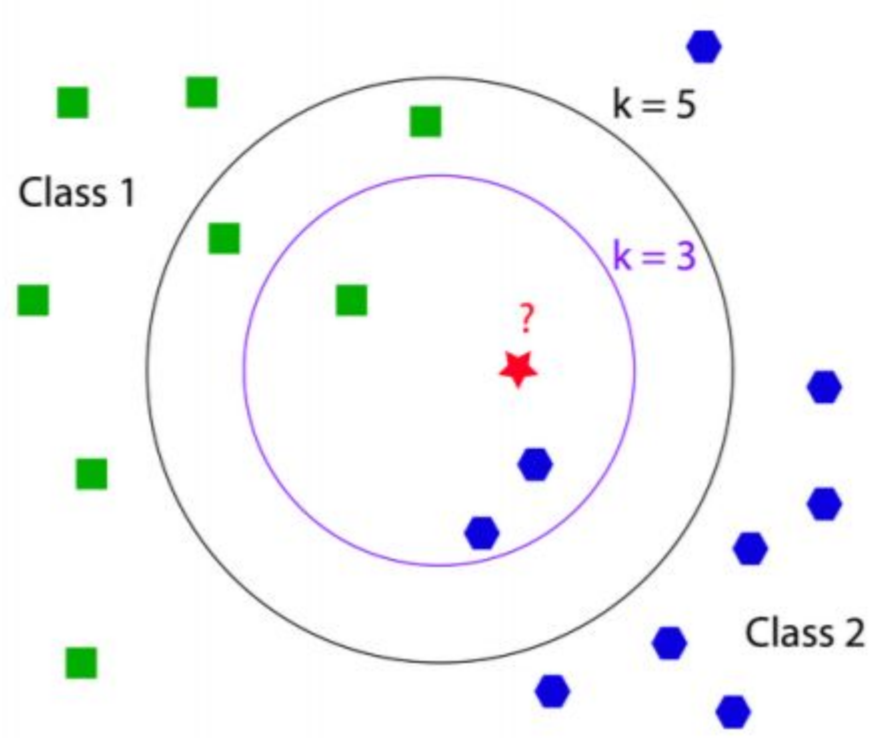
- Cosine

$$D(\mathbf{x},\mathbf{y}) = ||\mathbf{x}||||\mathbf{y}||cos(\theta)$$

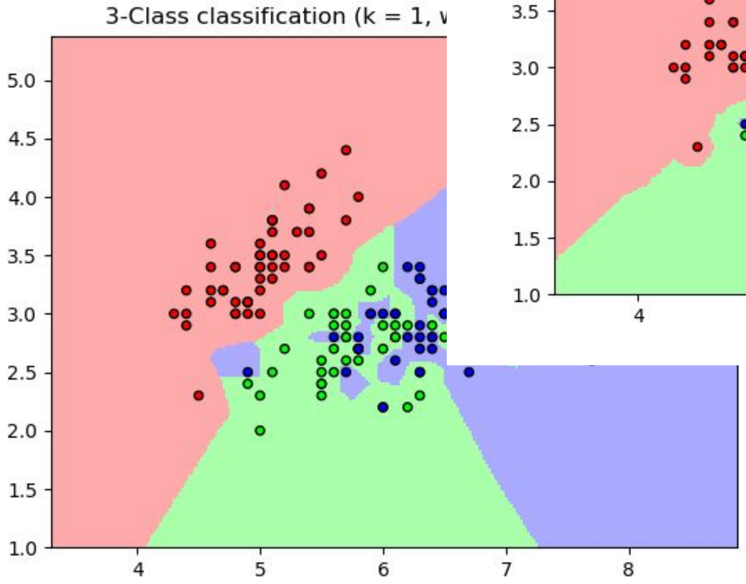
$$cos(\theta) = \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$$

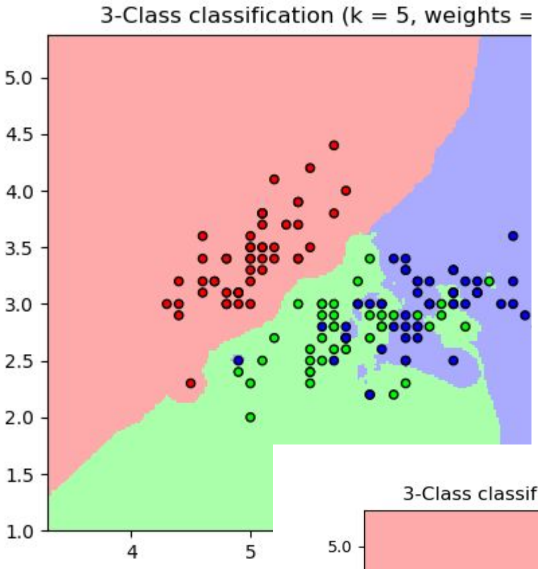
Choosing k

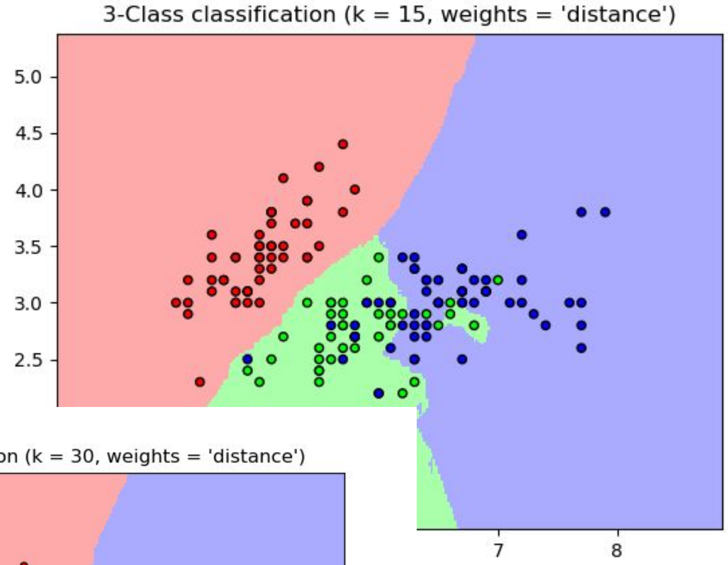
- k is a parameter that we need to tune.
- In the example, the example to be classified could be either a blue hexagon or a green square depends on how large k is.

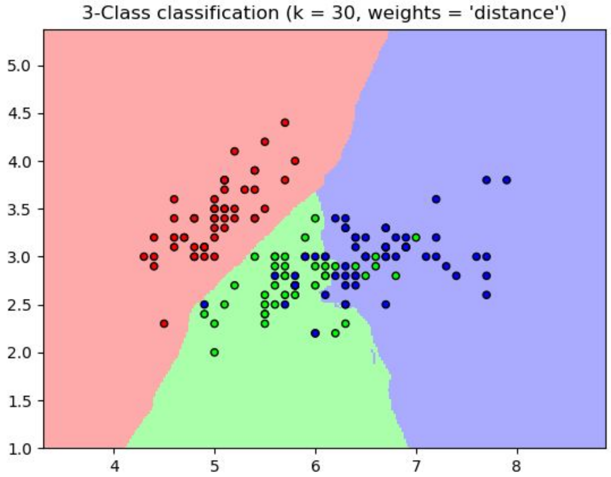


Choosing k









Adapted from [scikit-learn example]

Pros/Cons of kNN

- Intuitive and simple to understand
- Doesn't require training
- Easy to build online classifier: we can just classify new data as it comes

- Computationally intensive for large number of data points: fast for
- Sensitive to outliers
- Sensitive to un-balance dataset

Linear Classification

Linear Classification is not Linear Regression

Linear Classification: Given a training example, decide which of a finite number of classes it belongs to

- e.g. is this a dog, a cat, or a bird?

Linear Regression: Given a training example, estimate some continuous value

- e.g. what is the price of this house?

Linear Classification

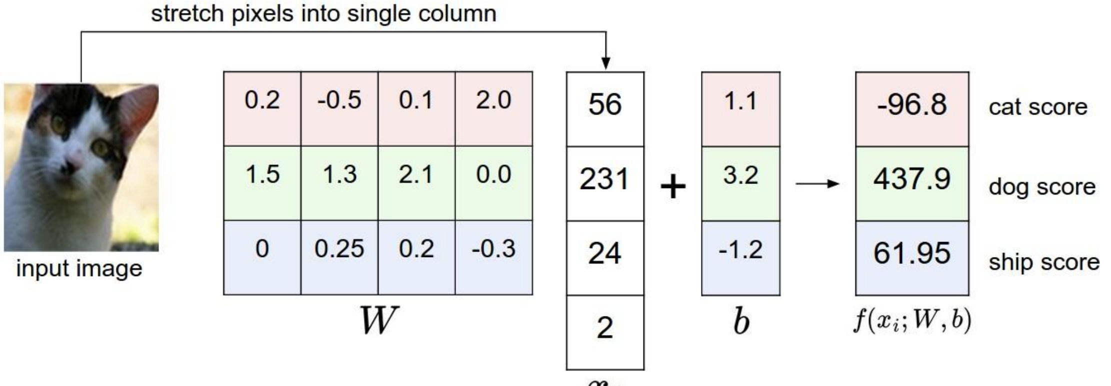
1. Input training data
2. Make predictions

$$f(x_i, W, b) = Wx_i + b$$

3. Compute loss
4. Update the model parameters

Linear Classification

- Represent input as a vector
- Matrix multiply to get class scores -- highest score wins



https://cs231n.github.io/linear-classify/

- W and b must be trained

Support Vector Machine (SVM)

(Multiclass) SVM = a loss function

Measures incorrectness of the prediction

1. Input training data
2. Make predictions
3. Compute loss

$$L = \frac{1}{N} \sum_{i} \sum_{j \neq y_i} \left[\max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + \Delta) \right]$$

4. Update the model parameters

Multiclass SVM

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta)$$

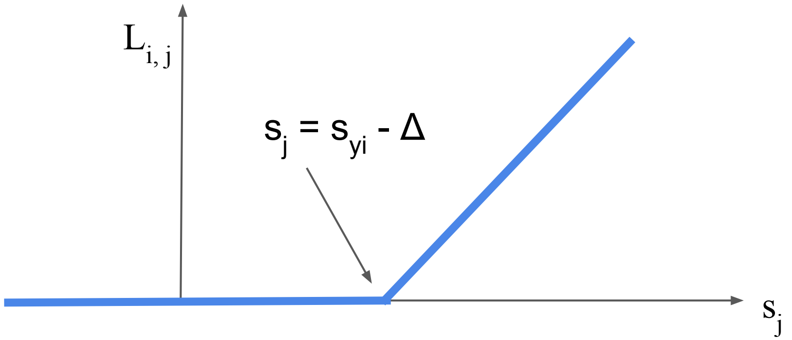
Loss for example i

All classes j except the class of this example i

Score for class j

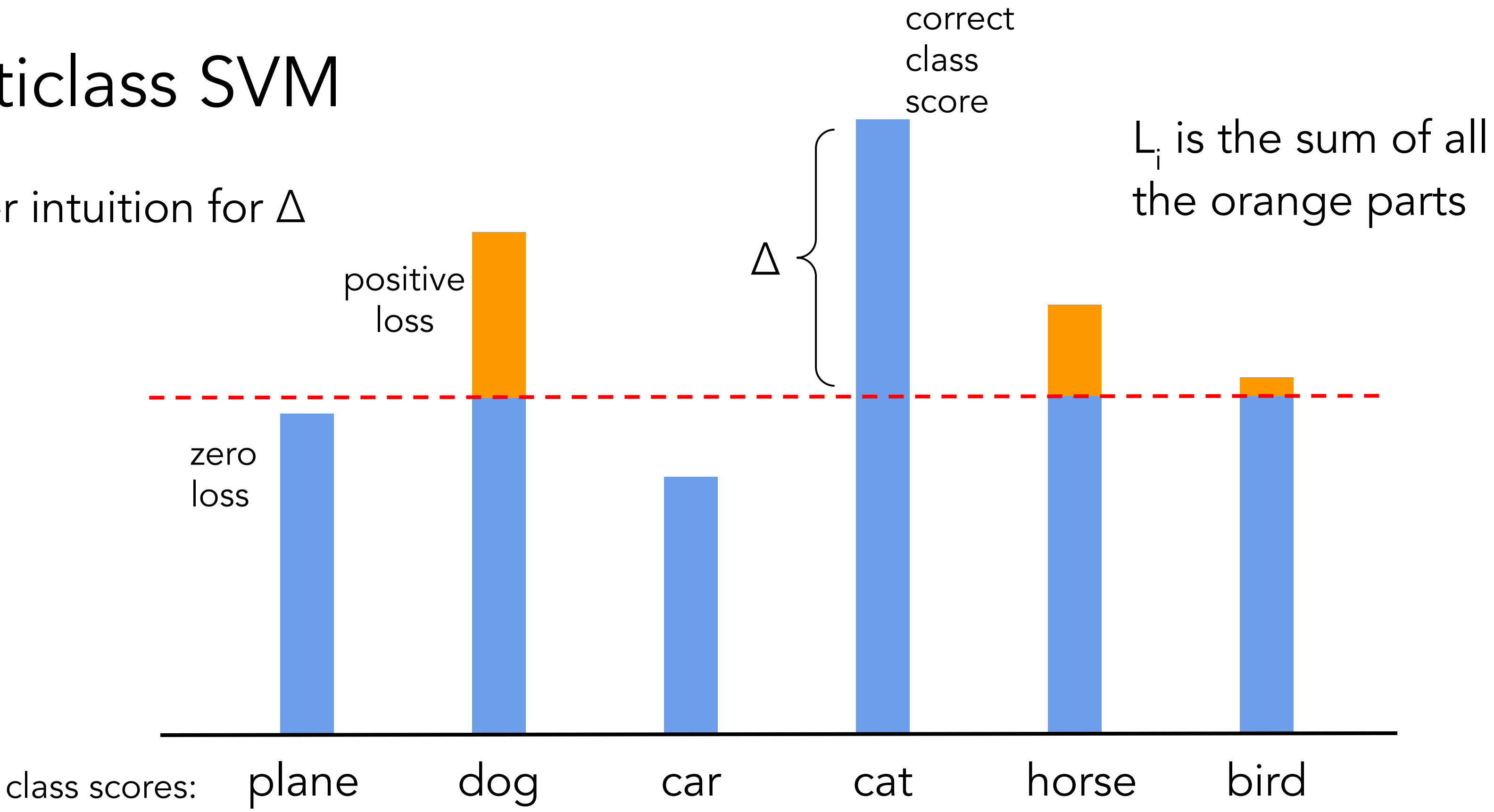
Score for true class y_i of example i

"Delta"



Multiclass SVM

Further intuition for Δ



Multiclass SVM

$$L = \frac{1}{N} \sum_i L_i$$

Total loss is average loss over all training examples

Multiclass SVM

4. Updating the model parameters…

- Update the parameters such that the loss is minimized

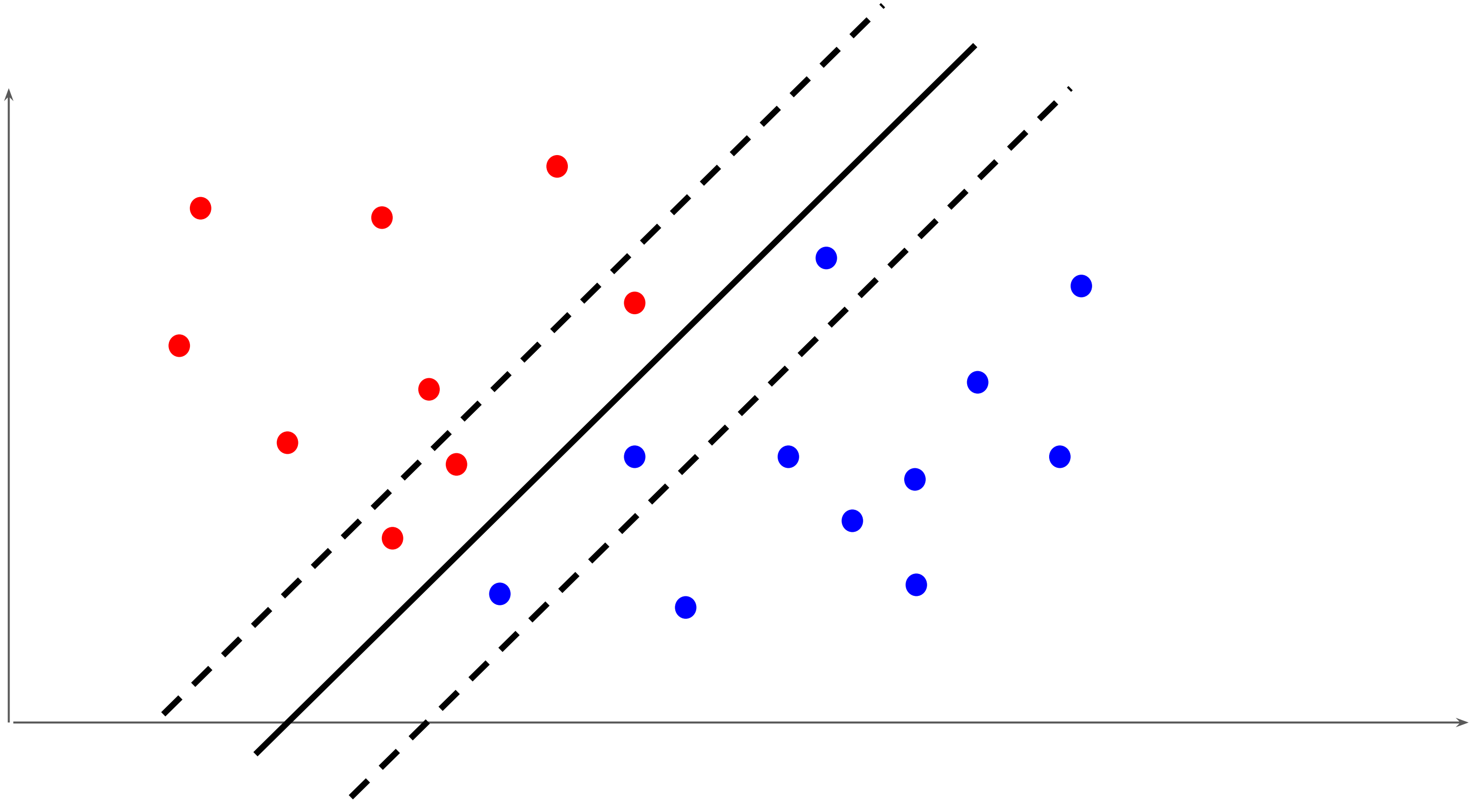
For weights of correct class:

$$\nabla_{w_{y_i}} L_i = -\left(\sum_{j \neq y_i} \mathbb{1}(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0)\right) x_i$$

For weights of incorrect classes:

$$\nabla_{w_j} L_i = \mathbb{1}(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0) x_i$$

Decision boundary



Pros/Cons of SVM

- Classification is fast
- Simple to implement

- Training is slow
- Is not sensitive to more complex relationships between variables

Upcoming Events

- Society and MI Discussion
  - Wed, February 20, 7pm – 8pm, MCS
- First Paper Discussion
  - Tue, February 19, TBD

Scaffolded kNN Exercise
http://bit.ly/scaffolded_kNN_example

Complete kNN:
http://bit.ly/complete_kNN_example

Regression Example Using SVM:
http://bit.ly/complete_SVM_example_2

1. You can choose to implement the algorithms with a scaffolded version or study the complete version of the code.
2. You would have access to both kNN and SVM solutions.
3. Challenge: extend the code into using one-versus-one multiclass SVM.