MSBD 5001 Tutorial 2

Jiacheng Xia

Outline

- Sklearn go-through
  - Models mentioned in lecture
- Reading relevant documentation
- Simulation: Comparing kNN and logistic regression
  - Demo code discussion

Sklearn

- “Scikit-learn”
- Simple and efficient tools for data mining and data analysis
- Open source
- Built-in datasets, algorithms, various examples

Models Mentioned in Lecture

- Logistic regression – sklearn.linear_model.SGDClassifier
- SVM – SVM
- kNN - sklearn.neighbors.KNeighborsClassifier
- Decision tree – tree
  - We will come to decision tree next tutorial

Reading Documentations

- Impolite “RT*M”, but manuals/documentations contain a lot of information.
  - Function definitions, variable meanings
  - Detailed usage
  - Simple examples
- Begin with example: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Reading Docs: Function Usage

class `sklearn.svm.` **SVC** (*C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None*) [source]

- Function name
- Parameter, after "=" is its default value
  - If you skip the value, it uses default.
- Detailed docs on each of the parameters

Reading docs: examples

- View examples to see how to use

```
>>> import numpy as np
>>> X = np.array([[-1, -1], [-2, -1], [1, 1], [2, 1]])
>>> y = np.array([1, 1, 2, 2])
>>> from sklearn.svm import SVC
>>> clf = SVC()
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> print(clf.predict([[-0.8, -1]]))
[1]
```

Demo 1

- Read a simple dataset from built-in library
- Split into training and testing set
- Train and test

Outline

- Sklearn go-through
  - Models mentioned in lecture
- Reading relevant documentation
- Simulation: Comparing kNN and logistic regression
  - Demo code discussion

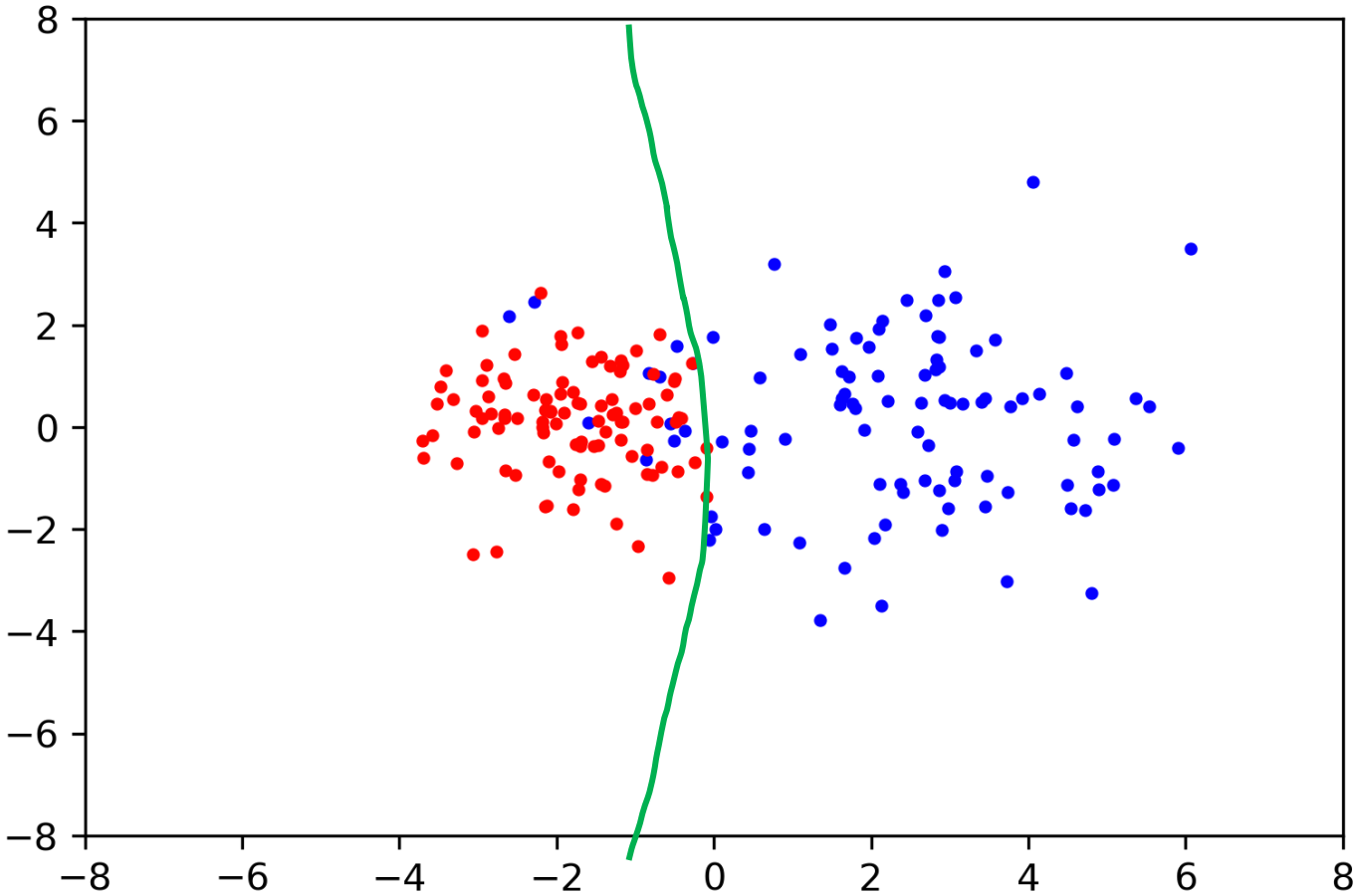
kNN vs Logistics Again

- When to use KNN, when to use logistics?
  - As discussed in lecture, depends on your data
- Consider 2 scenarios for 2-class classification problem*:
  - 1. Each class of training data are clustered among a center
  - 2. Each class of training data are clustered among a few randomly scattered points
  - Which one do you think works better?

* Please refer to sec. 2.3 of “Elements of Statistical Learning”

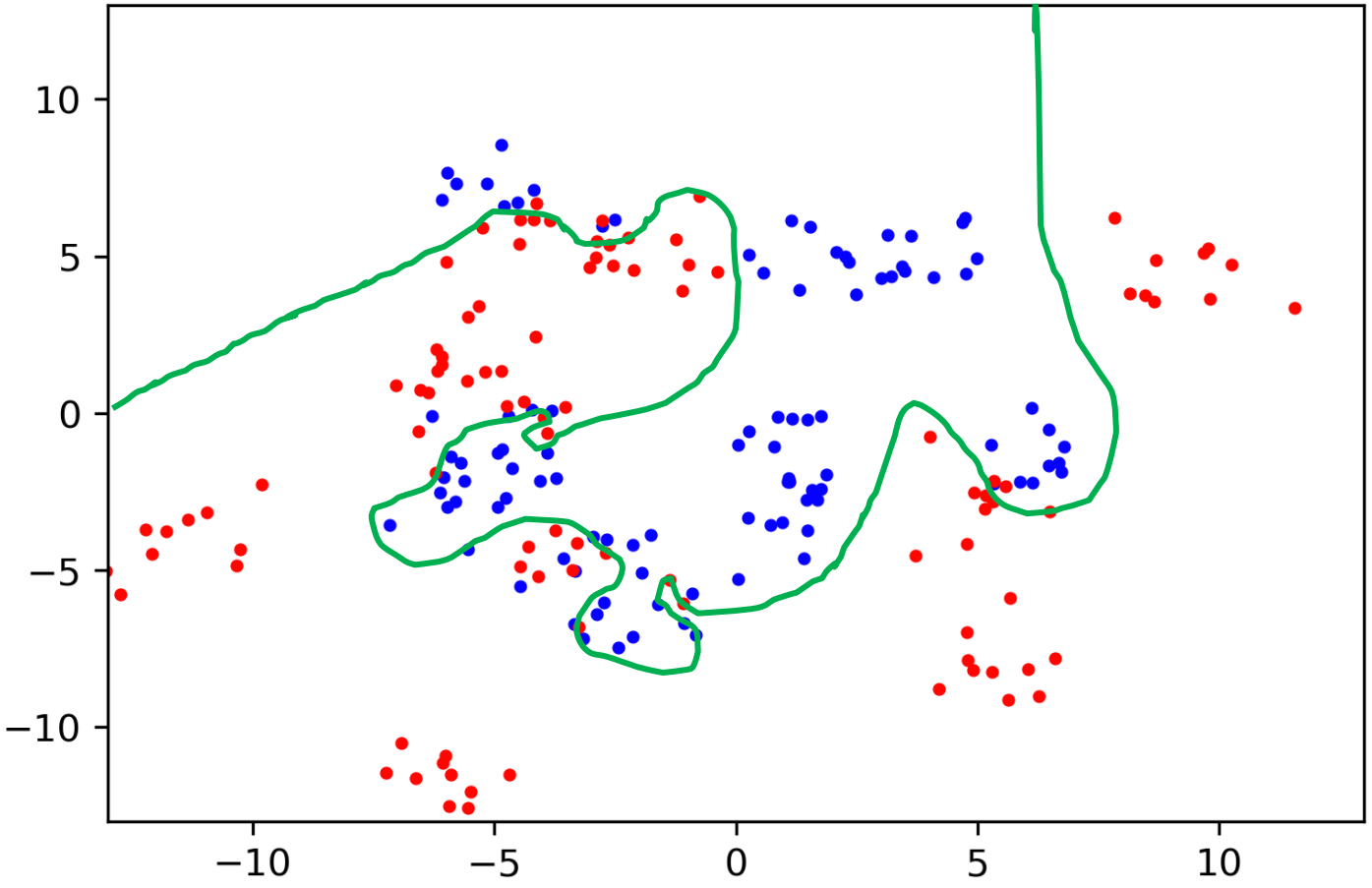
Scenario 1

We want a separator like this → similar to a straight line

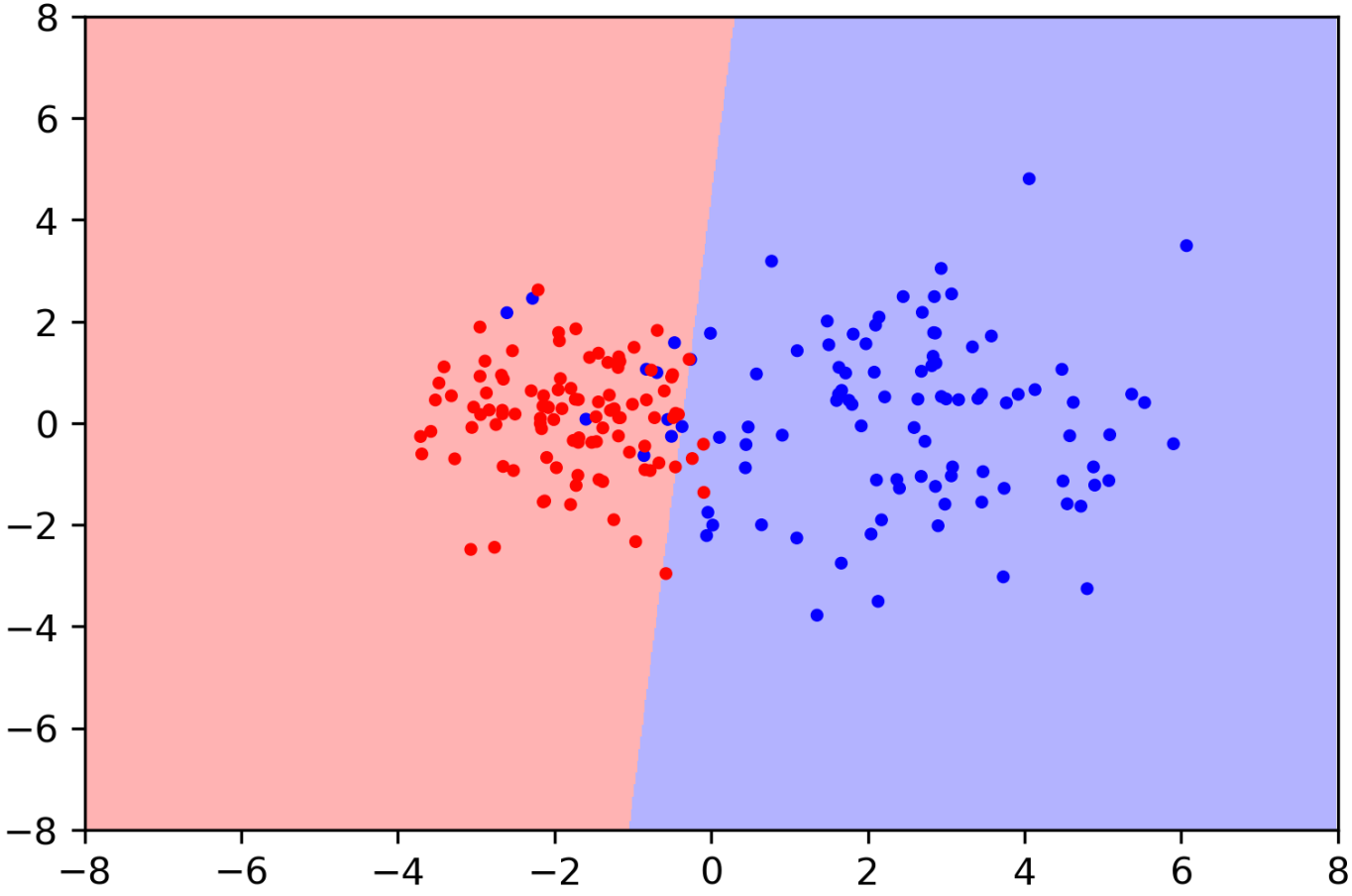


Scenario 2

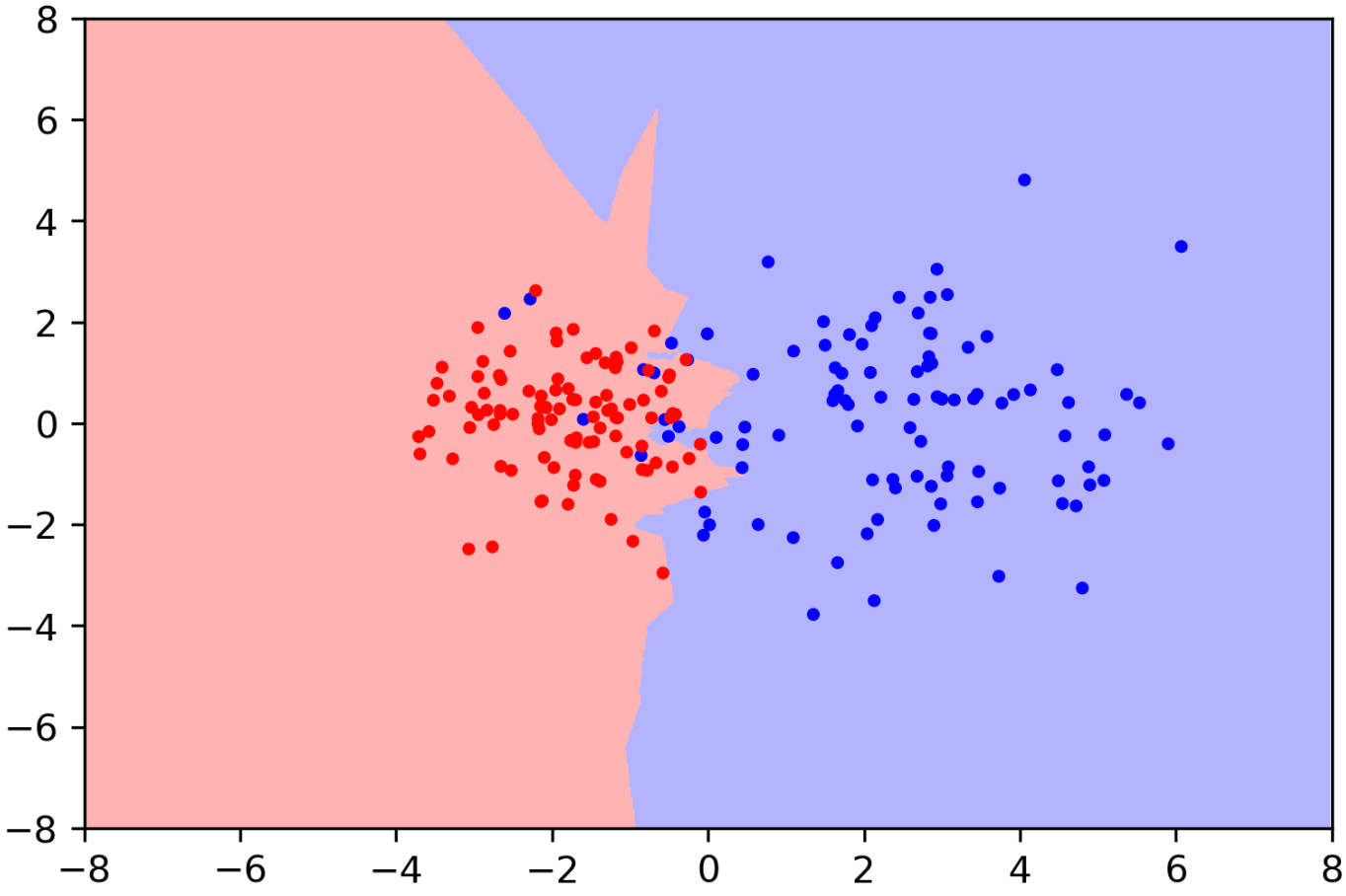
We want a separator like this → Not a straight line, though



Linear Classification in Scenario 1

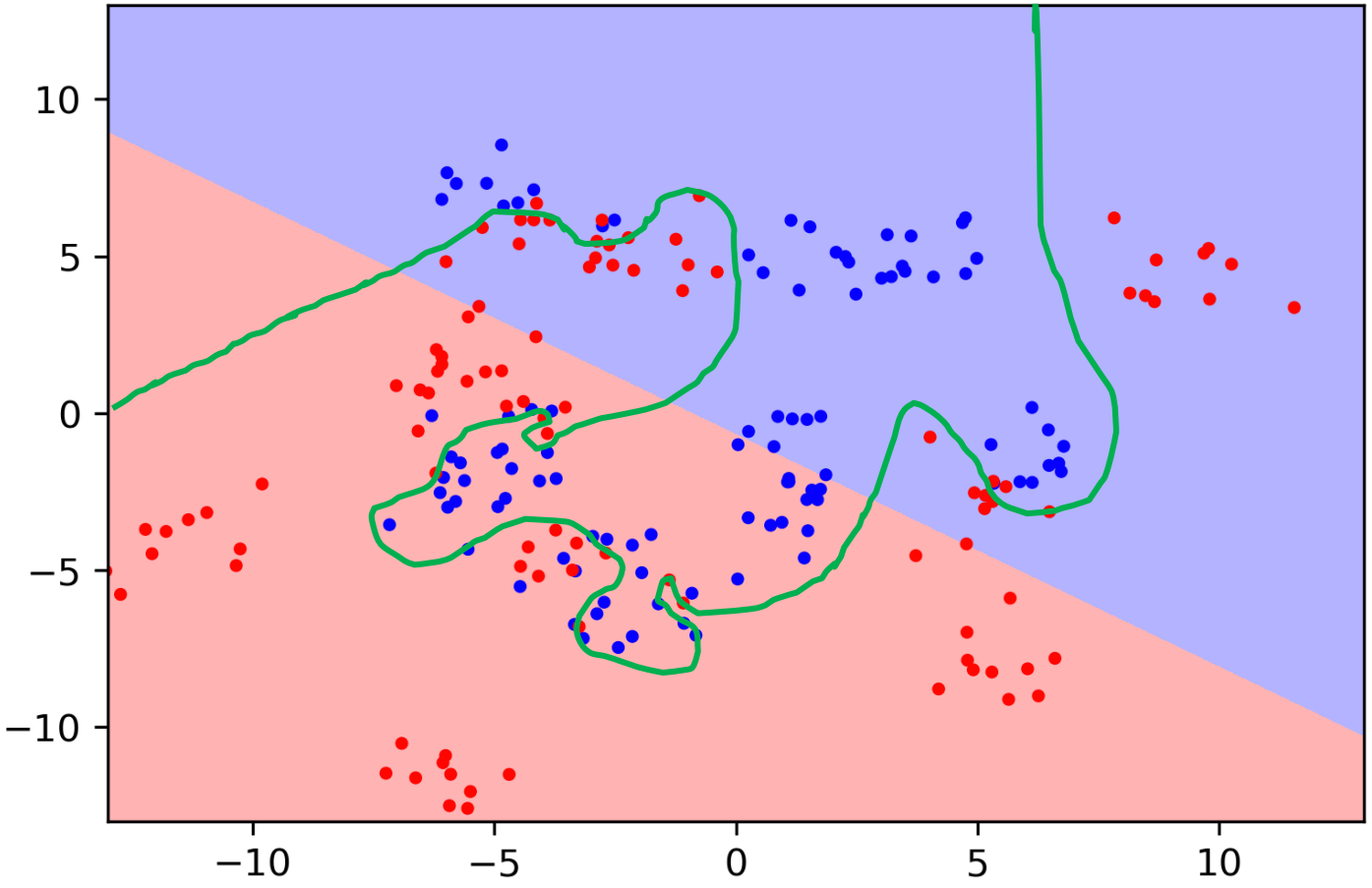


kNN in Scenario 1, k=5



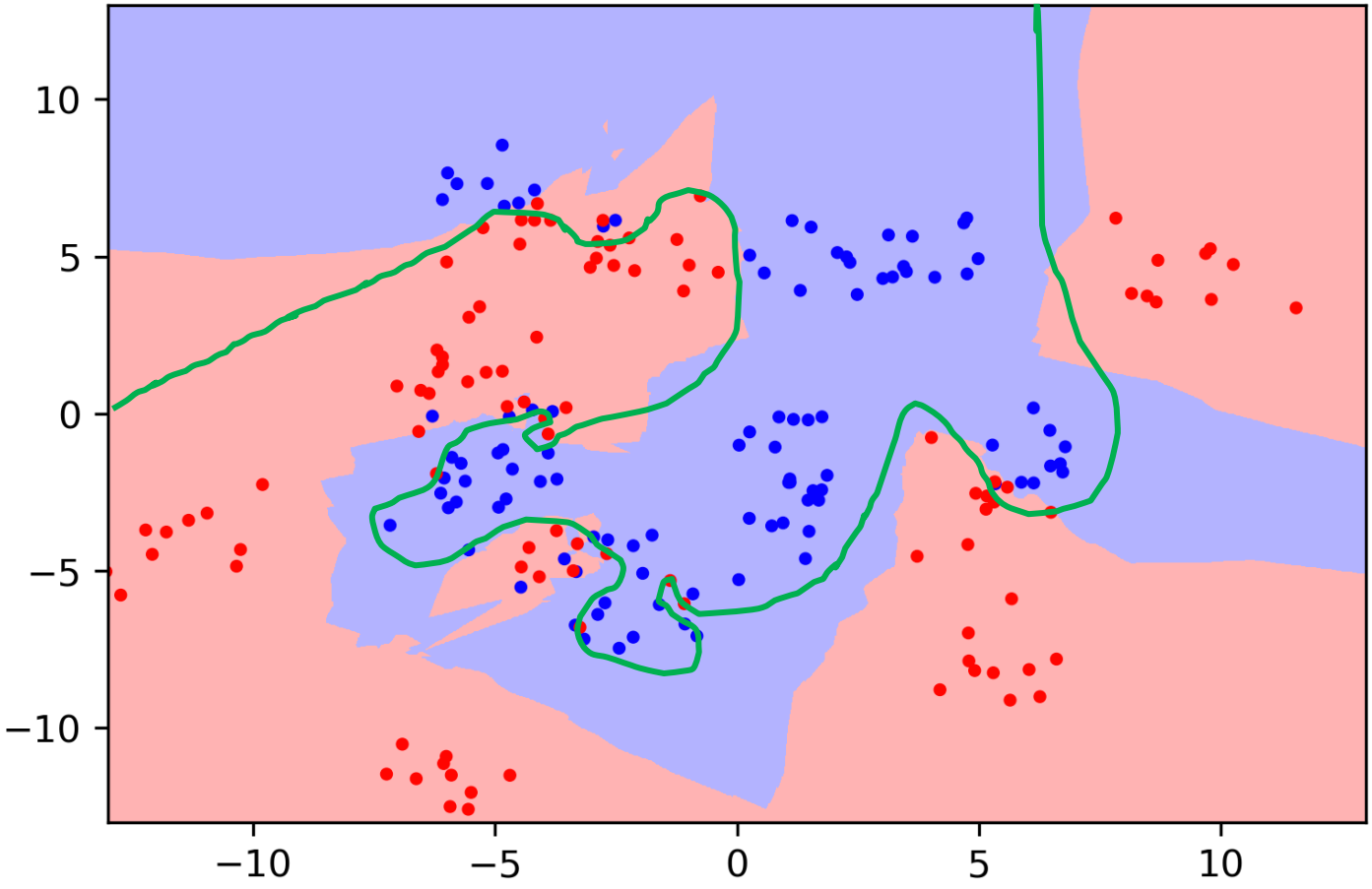
Linear Classification in Scenario 2

Use linear model when it is actually far from linear relationship?



kNN in Scenario 2, k=5

Similar results as visual view… How do you explain it though?



Demo 2 (Simulation 1&2)

- Generate random data
  - Random numbers
  - Set random seed to replay your result
- Visualizing your result with pyplot
- Code is available at GitHub