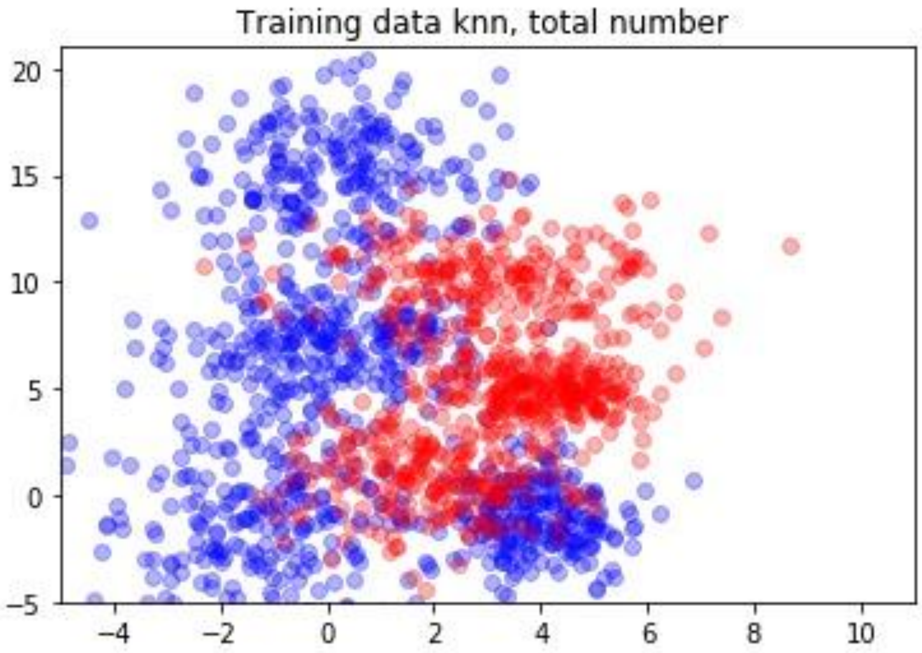
Mislabeling in training data

Agaton Fransson

Generating data

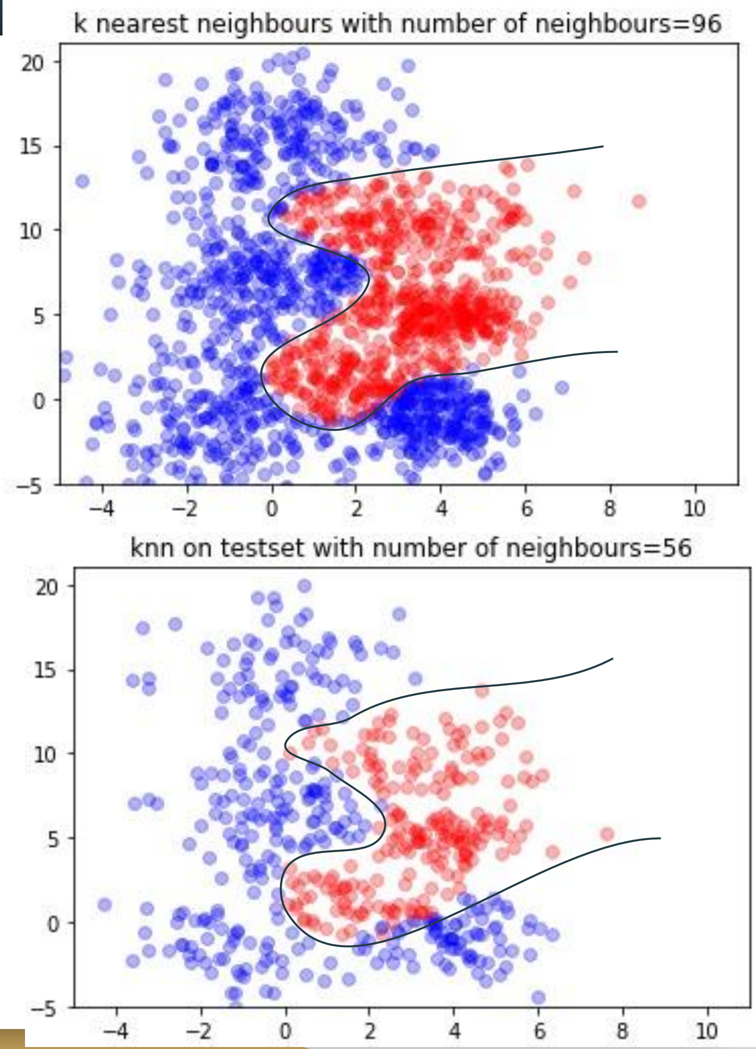
- Some bivariate normal distributions with different mean and variance, 300 points each
- Assign labels 1 (red) or 0 (blue) to the different data sets according to whim
- 3 distributions for 1 (red) and 4 distributions for 0 (blue)

Label	μ	Σ	Label	μ	Σ
0	(4,-1)	(1,0;0,2)	1	(4,5)	(1,0;0,1)
0	(0,7)	(2,0;0,2)	1	(3,10)	(3,0;0,3)
0	(0,15)	(2,0;0,6)	1	(2,1)	(2,0;0,3)
0	(-1,-1)	(2,0;0,3)			



k nearest neighbours

- Train on generated data
- Consistent “accuracy” around 87-88% for k between 10 and 100 on training data, 100% accuracy for 1 neighbour
- Accuracy=”Number of correct labelings” divided by ”total number of labels”
- Test accuracy on test set, generated by taking 25% of the points from each distribution
- Accuracy lower than 87-88% (around 82-86%) for k<10, settles at 87-88% for 10<k<100
- The borders take a very similar shape, they are drawn by hand



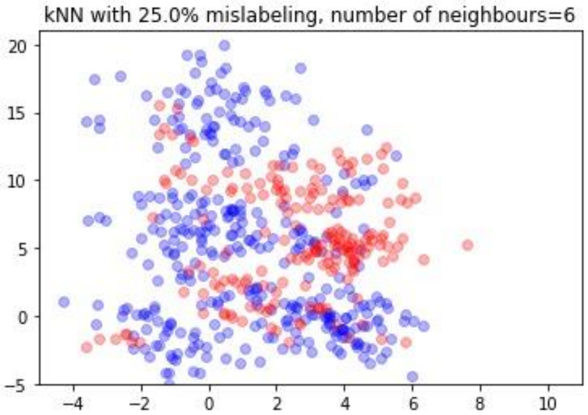
kNN with mislabeling

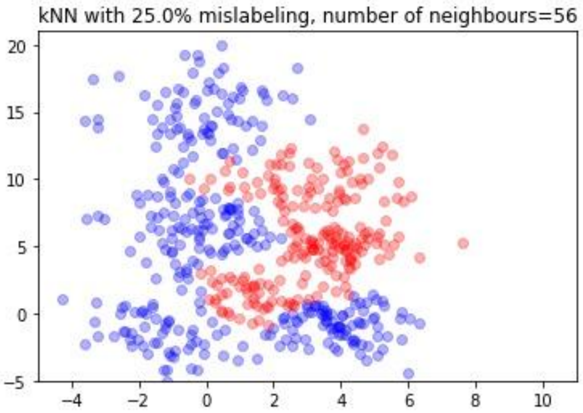
“Accuracy” consistent up to around 20% mislabeling, starts to go down for higher percentage.

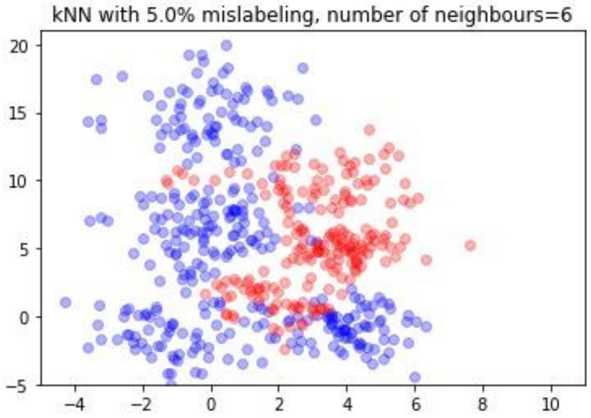
“Accuracy” increases as the number of neighbours increases, suggesting higher k gives resistance to mislabeling

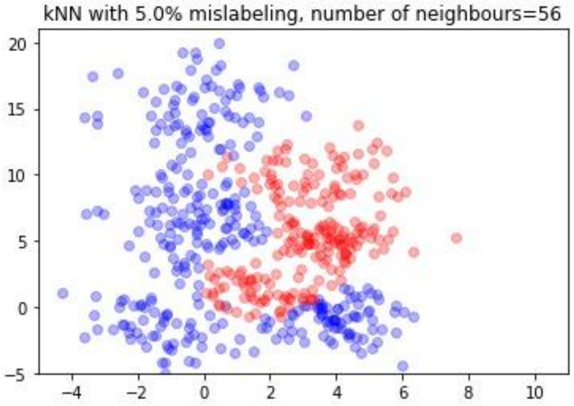
Accuracy goes from 75% to 87% for 25% error and from 85% to 87% for 5% mislabeling

The mislabeling is evenly distributed amongst the distributions of the training set


kNN with 25.0% mislabeling, number of neighbours=6


kNN with 25.0% mislabeling, number of neighbours=56

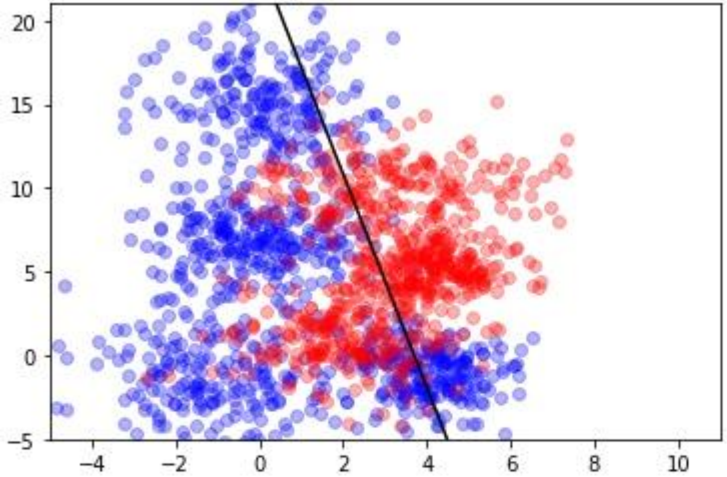

kNN with 5.0% mislabeling, number of neighbours=6


kNN with 5.0% mislabeling, number of neighbours=56

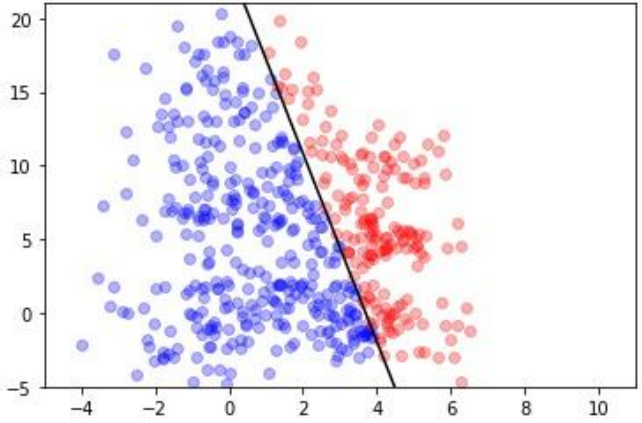
Linear model

- beta=[0.1352; 0.0221]
- Trained on 75% of generated data
- 69.0% accuracy
- Not same data as kNN but generated from the same distributions

Training data and linear model



Linear model with beta=[0.1352; 0.0212] trained on 75.0% of the available data



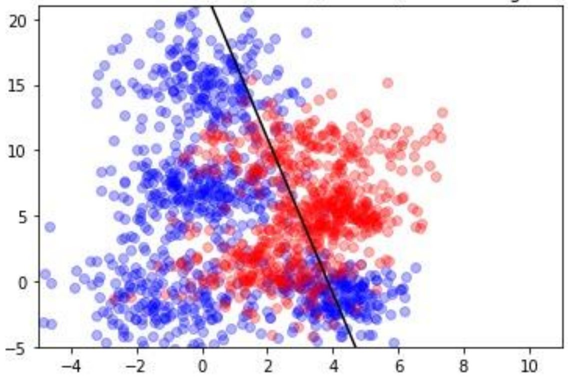
Linear model with error

69.3% accuracy for 5% error
70.5% accuracy for 20% error
69.5% accuracy for 33% error
63.8% accuracy for 50% error

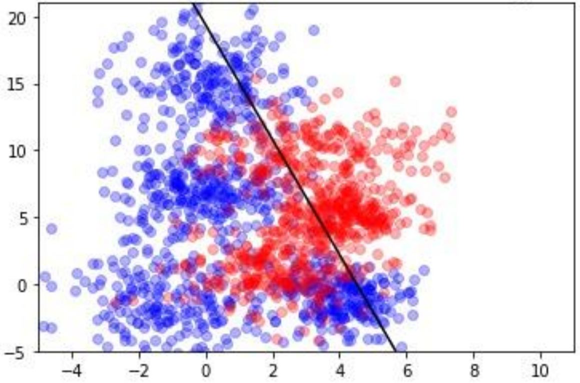
Higher accuracy might be because error in my code or that when increasing the slope, more red is mislabeled but more of the blue is correctly labeled, or that the distributions aren’t very well linearly separable

A line assigning all points to blue would give 57.1% accuracy

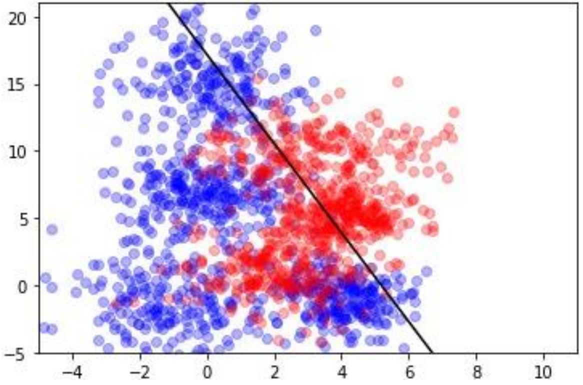
Linear model with beta=[0.1300; 0.0220], mislabeling =5.0%



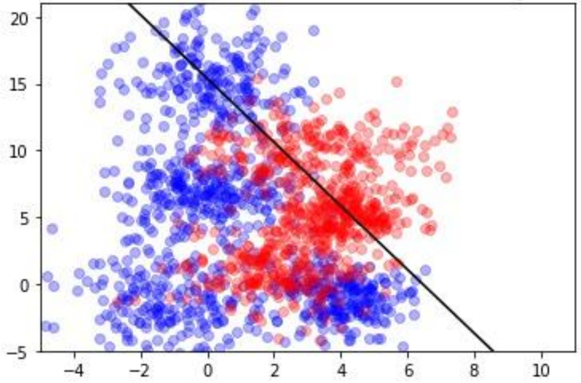
Linear model with beta=[0.1107; 0.0258], mislabeling =20.0%



Linear model with beta=[0.0964; 0.0291], mislabeling =33.0%



Linear model with beta=[0.0775; 0.0325], mislabeling =50.0%



Recovering wrong label

- kNN has 100% accuracy on its training set if it only checks 1 neighbour, accuracy decreases for increased k. Not true with test set, accuracy increased to a treshold with increasing k. Likely always true for overlapping distributions.
- The accuracy of the labels was low with mislabeled trainingset if the number of neighbours (=k) was low, it increased to almost the same accuracy without mislabeling if k was high.
- Accuracy somewhat constant for increasing mislabeling for linear model. Likely depends heavily on distributions since the linear model assumes linear classification line for distributions. Since my data wasn’t linearly separable the linear model works poorly in this case.