Introduction to Data Science

K-Nearest Neighbors

Gordon Anderson

KNN Characteristics

- Mainly used for classification tasks, but can be used for regression.
- Input: numeric values, or values for which a *distance metric* can be calculated.
- Classification output: a categorical value, or “class label”.
- Regression output: a numeric value- generally the average of the values of the k-nearest neighbors.

KNN Algorithm

- Beautiful as it is so simple!
- The value of k, the number of nearest neighbors, and a distance metric must be specified beforehand.
- Considered a “lazy” algorithm as it only computes when it has to.
- The algorithm (for classification) has training and classification phases.
    - The training data is of the form: $\langle y_i, \vec{X}_i \rangle$, where each *y* represents a class label, and *X* represents a vector of predictors, or as we often say in ML, “features”.
    - Note this is a supervised type of ML as examples of outcomes are provided.
    - The test data is of the form: $\langle \vec{X}_i \rangle$, just feature vectors. The algorithm outputs a vector of class labels that correspond to its predicted class labels for each feature vector.

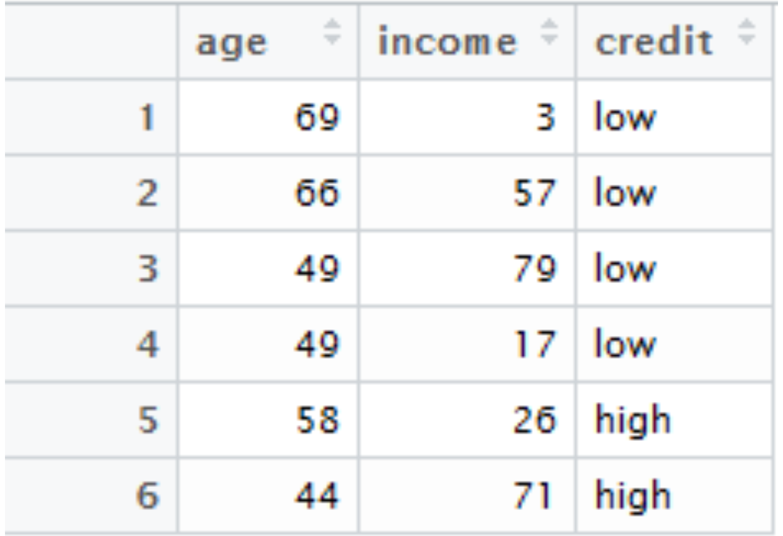
KNN Algorithm

- Training phase: just store the training data- that's it!
- Classification phase:
- For each feature vector X_{test} in the test data:
  - Find the distance from X_{test} to each vector in the training set:

    $$d_i = \text{distance}(X_{test}, X_{train})$$

  - Select the training data with the k shortest distances.
  - Collect the set of class labels from this subset.
  - The prediction is the most frequently occurring label in this subset (majority vote).

KNN Example



	age	income	credit
1	69	3	low
2	66	57	low
3	49	79	low
4	49	17	low
5	58	26	high
6	44	71	high

Predict credit rating based on age (years) and Income (thousands of dollars).

Training data:

  class label: credit {“low”, “yes”}

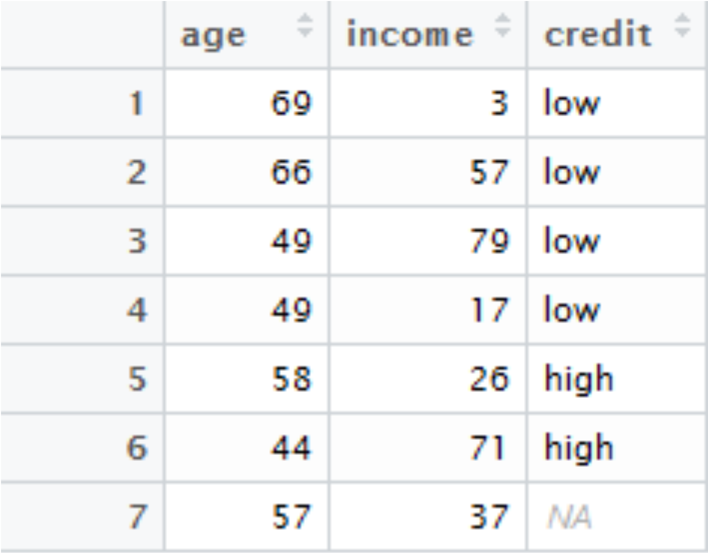
Features:

  age: numeric

  income: numeric

We need to specify k and the distance metric:
Let k=5, use Euclidean distance.

KNN Example



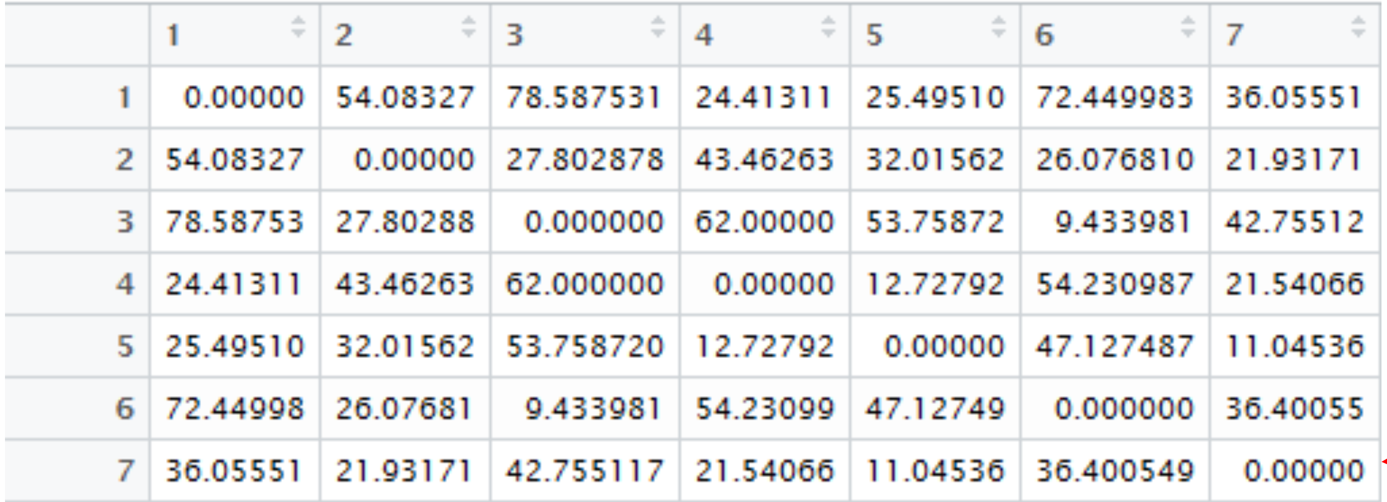
	age	income	credit
1	69	3	low
2	66	57	low
3	49	79	low
4	49	17	low
5	58	26	high
6	44	71	high
7	57	37	NA

Classification:

What label to assign the feature vector:
<57, 37>

Need to find the 5 nearest feature vectors using Euclidean distance, then look at the labels for those rows and then take majority vote.

KNN Example



	1	2	3	4	5	6	7
1	0.00000	54.08327	78.587531	24.41311	25.49510	72.449983	36.05551
2	54.08327	0.00000	27.802878	43.46263	32.01562	26.076810	21.93171
3	78.58753	27.80288	0.000000	62.00000	53.75872	9.433981	42.75512
4	24.41311	43.46263	62.000000	0.00000	12.72792	54.230987	21.54066
5	25.49510	32.01562	53.758720	12.72792	0.00000	47.127487	11.04536
6	72.44998	26.07681	9.433981	54.23099	47.12749	0.000000	36.40055
7	36.05551	21.93171	42.755117	21.54066	11.04536	36.400549	0.00000

The distance matrix calculated by Euclidean distance.

The vector we are predicting

The training vectors in order closest to farthest: 5, 4, 2, 1, 6, 3
These are the 5 nearest vectors: 5, 4, 2, 1, 6
Their labels are: “high”, “low”, “low”, “low”, “high”
The majority vote picks “low” as the predicted label for the vector <57, 37>
So, for age=57, income =$37,000 KNN predicts “low” credit rating.

KNN Issues

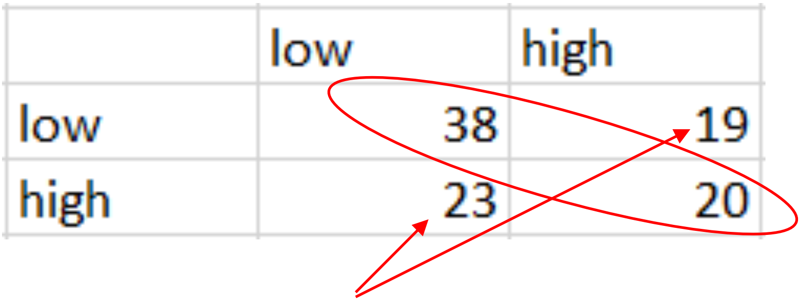
- The example predicted “low” since there were 3 “low”s and 2 “high”s in the 5 nearest neighbors. What happens if there is a tie? Toss a coin- also, don’t pick even k.
- The 3 versus 2 seems pretty close- could have gone the other way…
- This is a toy data set- so very small, but, could happen in a larger set.
- We hope that on average, the classifier gets it right more often than not (see following slide on evaluation metrics).
  - Can adjust the value of k and run again- this is almost always the case.
  - Can try a different distance metric.
  - Can weight the neighbors based on their distance.
- KNN is very flexible- can try a lot of things.

KNN classifier evaluation

- The simplest way to evaluate a classifier is to look at all possible outcomes of the true labels from the test set against the predicted labels from the classifier.
- This forms a table called a “confusion matrix”:

True labels

Predicted labels



	low	high
low	38	19
high	23	20

The correct classifications are on the diagonal

Misclassifications

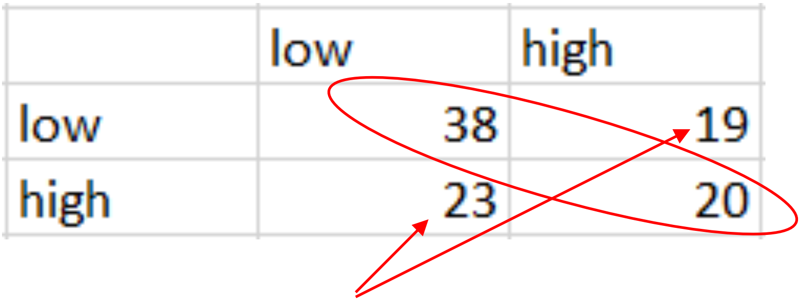
KNN classifier evaluation

- Common evaluation metrics:
- Misclassification rate: incorrect/total = 42/100 = .42 or 42%
- Correct classifications(accuracy): correct/total = 58/100 = .52 or 58%

Not great results- try different values of k.

True labels

Predicted labels



	low	high
low	38	19
high	23	20

Misclassifications

The correct classifications are on the diagonal

Total data size: N=100

Other classifier evaluation metrics:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

		Actual Values	
		Positive	Negative
Predicted Values	Positive	38	19
	Negative	23	20

True positive rate = TP/TP+FN = 38/61 = 62% ***Sensitivity (recall)***
False positive rate = FP/FP+TN = 19/39 = 49%
True negative rate = TN/TN+FP = 20/39 = 51% ***Specificity***
False negative rate = FN/TP+FN = 23/61 = 38%

More about these later on.

KNN Summary

- Very simple- can be adapted in many ways.
- Can use for regression and classification tasks.
- Wide range of distance metrics available.
- Can add weighting to nearest neighbors.
- Basic version- requires large storage for the data in training phase.
- Low use of computational resources.