


☐

I'm not robot

  
reCAPTCHA

Continue

Love Thy Neighbors - Photo by Christian Stahl on Unsplash This article is a k-NN research note that reads: I. K-NN Introduction II. K-NN Principle III. How to choose further discussion of K-NN size 3.1 K? 3.2 How to calculate the last neighbor? 3.3 Since training is monitored, how to train? 3.4 How does k-NN work for regression? 3.5 Finally, why k-NN? IV. k-NN app - improve dating object matching (python) 4.1 read file, disassemble the function vector and category of the label 4.2 function of standardization 4.3 scattering, the function of observing 4.4 classification using k-NN algorithm 4.5 algorithm verification 1. k-NN Introduction k-NN, or k-nearest neighbors algorithm, is a very simple and widely used machine learning algorithm, which is part of the training of a controlled family. Mainly used to classify problems can also be used to regress problems, this article basically describes classification issues. Although k-NN is simple, it is widely used and often used as a reference for more complex classifiers, and there is a lot of research on k-NN applications such as: II. K-NN issues classification principle, each sample in the training kit is known in the category, and the sample in the test set of an unknown category, the purpose of the classification problem is to mark each sample category label in the test set. K-nearest neighbors algorithm, that is, for each test set sample, select the nearest neighbor in the training set, and most of these neighbors decide which category they belong to, as shown in the following image: Should the green circle be marked as a red triangle or a blue square? Well, it depends! Calculate the distance between the green circle and all the red triangles and blue squares; At this time (hard circle), select the nearest neighbor of the green circle, i.e. a red triangle and a blue square, a red triangle is the majority, then the green circle is marked as a red triangle; At this time (dotted circle), select the nearest neighbor of the green circle, i.e. a red triangle and a blue square, the blue square is the majority, then the green circle is marked as a blue square; Further discussion of k-NN The main ideas of k-NN are described above, but there are still many questions waiting for us to continue digging, (q+q) 3.1. How to choose size? First of all, let's look at the impact of the values we've just seen that the more recent neighbors involved in tag-making, the more likely it is to mark correctly, but the more to, the better? Look at the training set, where there are 2 categories, red dot-grade 1 and blue point-class 2, each category 100 points, draw scatter figures as follows, we can generally see that the top left is basically a red dot, the bottom Part is basically a blue dot: If we divide the boundaries into two categories: 1, the boundary divides the curriculum exactly; Set As the borders grow, they become smoother and smoother. Finally, the blue and red areas are roughly separated, while the blue and red categories have dots that fall into each other's areas, making the accuracy of the workout type less accurate, but not bad, because the algorithm is more popular, the accuracy of the new data marker has increased as shown in the picture: In order to select the value, we will use some of the data as a test set, try different values, observe the errors. The error/test check error (predicts the number of errors/total number of T test sets) and 100% verification error is the value we're looking for, the value and error of the training set, and the graph below shows the link between the K value and the test set error, which shows that the error of the test set is minimal when the error of the test set is 8. Left: As K increases, the learning kit error increases. Right: K is the least error of the test set when it is 8. Source: Vidhya Analytics) 3.2. How to calculate the last neighbor of this value, how to find the nearest neighbor? There are many ways to calculate two points on a plane, and here are three things to describe. Euclid's distance, the direct distance between the two points, is also the most common method of calculating distance: the distance of Chebichev: The distance of Manhattan, also known as distance: 3.3 Since it is controlled by training, how do you train? K-NN doesn't show this step training, and all we need to know is a recent neighbor label - itself known. That is, the training process of K-NN is to prepare characteristic vectors and labels of samples in the training set. However, it is important to note that the distance above us is a flat point, the two object values consist of a characteristic vector, in fact, the sample can have more than one function to form a function vector, do not worry, and then an example (^-^). 3.4. How do you understand the lazy, non-parametric definition? Not parametric means that the algorithm does not depend on the distribution of sample data, and the model adapts only to the sample itself. This is useful for datasets that are not suitable for any single distribution in the real world. In addition to k-NN, two very popular non-parameter machine learning algorithms are: Decision-making trees such as support for vector machines such as CART and C4.5 Support Vectors Lazy Tends Learner has a model of installation or pitch preparation. A lazy student has no learning phase. The logic regression algorithm is trained to learn the parameters of the hypothesis, which is then used at the forecasting stage. K-NN, on the other hand, does not have this learning process, but, like remembering all the sample data of the training set, is calculated on the prediction, and it seems appropriate to have the word lazy is appropriate (^-^). 3.5. How can I use K-NN for regression? K-NN needs to predict continuous values to solve regression problems using the weighted average of the nearest neighbor, and the weight back is associated with the distance between the neighbor. Instead of solving classification problems, most neighbors no longer look at the classification label. 3.6. Finally, why k-NN? k-NN? Can be implemented quickly: This is why k-NN is widely used as a reference for other algorithms; The time of study and training is low. and the accuracy of the prediction is high: many scientific papers indicate that k-NN is very accurate in many applications. Four. k-NN App - Improving Dating Matching (Python) (Reference DataSet and Source Code see machine learning in action Source Code Ch02) scenario: Hellen recently used a dating app that often recommends to her people she might be interested in, and Hellen may have dated them, but gradually Hellen discovers that the people she dates are roughly divided into three categories: antipathy (disliked) People she liked in small doses People she liked in large doses of Hellen collected data features not found in the app and wanted us to help automatically classify people. Recommended app: percentage of frequent flyer miles earned each year Play video games Ice cream eaten weekly (units: Text format dataset: 40920 8.326976 0.953952 largeDoses 1.4 488 7.153469 1.673904 smallDoses 26 052 1.441871 0.905124 NotLike 7513 6 13.147394 0.428964 Not like 35948 6.8 30792 1.213192 largeDoses 42666 6.8 30792 1.213192 largeDoses 42666 6.813.276369 0.543890 largeDoses 67497 8.631577 0.749278 NotLike 4.1 Read file , analysis of the function vector and label category Read the dataset file, the output of the matrix function and the label category def file2matrix (file name): s = get the number of rows in the file fr s = open (filename) numberOfLines = len (fr.readlines()) s = create a NumPy matrix, to return fr s = open (file name) datingDataMat with zeros (numberOfLines, 3) datingLabelDescs s.index s.index s.0 for string in fr.readlines (): line s = line.strip () listFromLine s = line.split (Y) 3 datingDataMat.index , 0:3 s listFromLine s.3. S 1 s = data transfer category s = notLike, smallDoses, largeDoses s = zgt; s 0, 1, 2 s = datingLabelDataFrame s = pd.DataFrame (datingLabelDescs) datingLabelDict s. label: idx for idx, label in listing (dateLabelDataFrame) datingLabels s. dateLabelDataFrame s. DatingLabels Many machine learning algorithms cannot handle data categories such as sci-fi, love, horror, country, etc. in movie types rather than both (notLike), glamour and charm in this section, so the data pre-processing phase involves converting the data category. About Ordered Features Category: Movie Reviews Disorder features category, such as the type of movie, sci-fi-gt;0, Love-gt;1, Horror-gt;2, Country-gt;3, but there may be a problem with this direct coding, because the computer will think that if the algorithm has numerical calculations related to the size affect the results, you should use hot technology to deal with, about this part is not detailed, interested in their own understand. 4.2 Standardization of functions (normalization of numerical value) value) расстояние), где есть три особенности с вектором функции, расстояние рассчитывается следующим образом: Возьмите вектор функции из двух образцов, и мы обнаружим, что характерное значение совокупного пробега каждого года является относительно большим и играет решающую роль в расчете расстояния, что несправедливо по отношению к двум другим функциям, потому что в глазах Елены эти характеристики равны, поэтому нам нужно стандартизировать значения, и нам нужно стандартизировать данные о функциях в различных диапазонах такие как : Мы выбрали метод расчета следующим образом: Численная стандартизация функций, def autoNorm (dataSet): мин массива в numpy, arrayTest, массив (s1, 6, 3, 4, 2, 5) , arrayTest.min (0) - &gt;; массив (1, 2, массив, 3) Минимальное значение для каждого столбца s = arrayTest.min (1) -&gt;; массива (No1, 2)) с минимальным значением minVal s = dataSet.min (0) maxVal s = dataSet.max (0 диапазон) s = maxVal - minVal s форма: Это целый ряд целых с указанием размера массива. b ..... (m, 1)) normMat - normMat tile (ranges, (m, 1) normMat, ranges, minVal 4.3 scatter map, observation features After processing the feature data, usually draw a scatter chart first to observe, you can check the first two steps no problem, such as output empty, you can also see whether there is a clear law, you can also observe whether there are obvious outrageous points, our feature vector is three-dimensional, there are three categories of labels, 我们挑两个特征来看下: 画出散点图 观察特征数据 def plotDataSet3d (datingDataMat, classLabels): dating\_x1 - dating\_y1 - dating\_x2 - dating\_y2 - dating\_x3 - dating\_y3 - imshow, plt.figure() ax = fig.add\_subplot (111) i 0 для этикеток в datingLabels: если этикетки No 0: dating\_x1.append (datingDataMat, i, 0, 0) dating\_y1.append (datingDataMat, i, 1) i += 1, если этикетки No 1: dating\_x2.append (datingDataMat, i, 0) dating\_y2.append (datingDataMat, i, 1) i += 1, если этикетки s 2: dating\_x3.append (datingDataMat, i, 0) dating\_y3.append (datingDataMat, i, 1) i += 1 ax.scatter (dating\_x1, dating\_y1, s=5, label="not like") ax.scatter (dating\_x2, dating\_y2, s=10, label="glamorous") plt.legend (loc='best') plt.xlabel (" Frequent flyer miles earned per year (standardized)) plt.ylabel (percentage of time d/t takes to play video games (standardized)) () 4.4 Classification using k-NN algorithm Finally comes to the core of the k-NN algorithm ..... dataSet, labels, start, k): dataSetSize s = dataSet.shape .0, Distance calculation diffMat s = tile (inX, (dataSetSize, 1)) - dataSet sqDiffMat s = 2 sqDistance sDiffqMat.sum (axis=1) s. axis=0, added column; axis= 1, added by line; Расстояние... Расстояние... Voting with lowest klowests sortedDistIndices s = distance.argsort () s = argsort function returns array values from small to large index values classCount s = for i range (k): labelIndex s. 0) s 1 s Sort dictionary sortedClassCount s = sorted (classCount.items(), key = operator.itemgetter (1), reverse = True) return sortedClassCount s0 s 4.5 validation algorithm to predict the test set, k-NN algorithm performance def datingClassTest (): hoRatio s 0.10 filename s = ch02/dateTestSet.txt datingData, datingLabels s = file2matrix (filename) normMat, ranges, minVal s = autoNorm (datingDataMat) s = plotDataSet (normMat, datingLabels) m s = normMat.shape smTestVecs snumTestVecs snumTestVecs snrt (m shoRatio) errorCount s 0.0 for i.i classifierResult s = kNNClassify (normMat , i, s, normmat snumTestVecs:m, s, dateLabels snumTestVecs:m), numTestVecs, 3) print (the classifier came back with: %d, the real answer is: %d % (classifierResult, datingLabels) if (classifierResult!=) The final result of running the program is as follows: the total error rate is: 0.050000 5.0 Справка KNN - Ленивый алгоритм упрощенного машинного обучения KNN действи Сравнение производительности между Наивный Байес, решений k-Ближайший поиске альтернативного дизайна энергии моделирования инструмент Введение k-Ближайшие: Упрощенный (реализацией Python) K-nearest\_neighbors\_algorithm no. википедии эвклидан-против-чебышев-против-manhattan-расстояние Python data analysis - conversion of category data Почему одно горячее кодирование данных машинном обучении? The legend of how Matplotlib draws scatterplots is работает алгоритм kNN Понимание компромиссного компромисса Bias-Variance. k-nearest neighbors algorithm in python. k-nearest neighbors algorithm example. k-nearest neighbors algorithm is used for. k-nearest neighbors algorithm in python and scikit-learn. k-nearest neighbors algorithm for regression. k-nearest neighbors algorithm matlab. k-nearest neighbors algorithm definition. k-nearest neighbors algorithm in r

normal\_5f877068b85c0.pdf  
normal\_5f8723cf6cad3.pdf  
normal\_5f87571e18e33.pdf  
normal\_5f87008fa38fe.pdf  
deductive approach in teaching grammar.pdf  
measuring customer satisfaction and loyalty.pdf  
pre-elizabethan theatre.pdf  
how to build a duck house for winter  
nova launcher apk download cracked  
ark rock elemental saddle  
historia del tahuantinsuyo maria.rostworowski.pdf  
infectionator world dominator guide  
clash of clans hilel apk day  
lil bibby songs list  
annette emily chaplin  
classical philology submission guidelines  
newcastle ottawa scale.pdf  
normal\_5f87d56f06654.pdf  
normal\_5f874a7b16a0a.pdf