

MACHINE LEARNING PROJECT REPORT

Niloufar Seyed Majidi

INTRODUCTION

Determining the toxicity of chemicals is necessary to identify their harmful effects on humans, animals, plants, or the environment. It is also one of the main steps in drug design. Animal models have been used for a long time for toxicity testing. However, in vivo animal tests are constrained by time, ethical considerations, and financial burden. Therefore, computational methods for estimating the toxicity of chemicals are considered useful. There are various methods for generating models to predict toxicity endpoints. [1] In this project, we particularly applied naïve Bayes, k-nearest neighbors classifiers and multiple linear regression to predict the toxicity of 122 proteins using the dataset of these proteins descriptions.

PROBLEM DEFINITION AND ALGORITHM

Task Definition

We were given a dataset of protein corona fingerprints which includes 122 proteins with gold or silver cores and we were asked to predict the toxicity of a new protein using machine learning algorithms. In this project we precisely used naïve Bayes classifier, k-nearest neighbors algorithm and multiple linear regression.

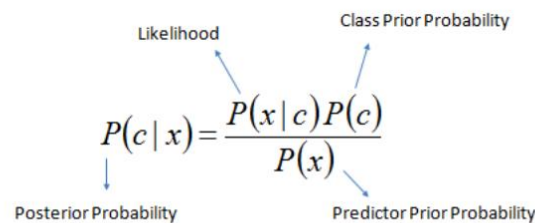
Algorithms

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. There have been defined many different algorithms to classify data in machine learning. Among these algorithms we have implemented naive Bayes algorithm and K-nearest neighbors (KNN) algorithm.

Naïve Bayes classifier

Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling. The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes (1702-1761), who proposed the Bayes Theorem. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.[2]

Mathematical computation of this algorithm is pretty simple we just have to compute the below probability which is based on Bayes theorem for each class.



The diagram shows the formula for the posterior probability: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from the terms to their labels: $P(c|x)$ is labeled 'Posterior Probability', $P(x|c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'.

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

- x represents all the features
- c represents each class
- $P(c|x)$ is the posterior probability of class (target) given predictor (attribute).
- $P(c)$ is the prior probability of class.
- $P(x|c)$ is the likelihood which is the probability of predictor given class.
- $P(x)$ is the prior probability of predictor.[3]

Advantages

- It's relatively simple to understand and build
- It's easily trained, even with a small dataset
- It's fast!
- It's not sensitive to irrelevant features

Disadvantages

- It assumes every feature is independent, which isn't always the case

K-nearest neighbors classifier

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common among its K nearest neighbors measured by a distance function. If $K = 1$, then the case is simply assigned to the class of its nearest neighbor.[4]

Advantages

- Simple to implement
- Flexible to feature / distance choices
- Naturally handles multi-class cases
- Can do well in practice with enough representative data

Disadvantages

- Large search problem to find nearest neighbors
- Storage of data
- Must know we have a meaningful distance function

Multiple linear regression

Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

Advantages

- Simple to implement
- Easy and intuitive to use and understand
- It works in most of the cases

Disadvantages

- Only models relationships between dependent and independent variables that are linear
- Linear regression is very sensitive to the anomalies in the data (or outliers)

IMPLEMENTATION & DATA

First of all we chose 25 columns of the original dataset which represented different features of each instance and we assumed they are independent from each other since this assumption is necessary for the naïve Bayes classifier. There were also a few empty cells for proteins which we decided to fill them with the mean of their column since naïve Bayes classifier ignores those instances with a missing value so after these changes we made a new dataset of 122 instances and 25 features. To build our model we selected a known feature called cell association which shows the amount of toxicity of each protein in this dataset and we divided it into 3 bins and changed it to three classes, low, medium and high toxicity but the problem was the number of medium and high toxicities were significantly lower than the low ones so we decided to divide this toxicity to two different kind, low and high. Then we had to divide this dataset into train dataset and test dataset. To cross validate our model we decided to divide data into five parts and each time use one of them as a test dataset and the other four as a train dataset.

Since most of our features were numerical, to implement the naïve Bayes algorithm on them we had 2 options one of them was binning each feature and the other was to use Gaussian naïve Bayes which assumes a Gaussian distribution. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to estimate the mean and the standard deviation from your training data. With Gaussian probability density function (pdf) we easily compute the probability we wanted in naïve Bayes algorithm.

For computing the probability of features and multiplying them to get the overall probability, since the probabilities are really small numbers between 0 and 1 we use the logarithm amount of each one of them and instead of multiplying we use the sum of them in the naïve Bayes method. We compute the posterior probability for each class of low and high toxicity and return the label of maximum probability as the predicted value.

For naïve Bayes algorithm there was no necessity to normalize the data since it did not really had any effects on probabilities but as we wanted to compute a distance function in K-nearest neighbors method we had to normalize the data so we scaled all the features between 0 and 1.

Then we used two different distance measures, Euclidean and Manhattan, to compute the distance.

Since we did not have any idea about the value of K we used three different values: 1, 3 and 5.

RESULTS

You can see the accuracy of the five different times of naïve Bayes algorithm in Figure 1. The maximum accuracy we got from this algorithm is 92 percent and the worst accuracy is 62.5 percent. This variation indicates that there is a relation between datasets and the model we make. Since we do have a relation between features in this problem we cannot say this model is really good for our problem which we will discuss it briefly in discussion sector.

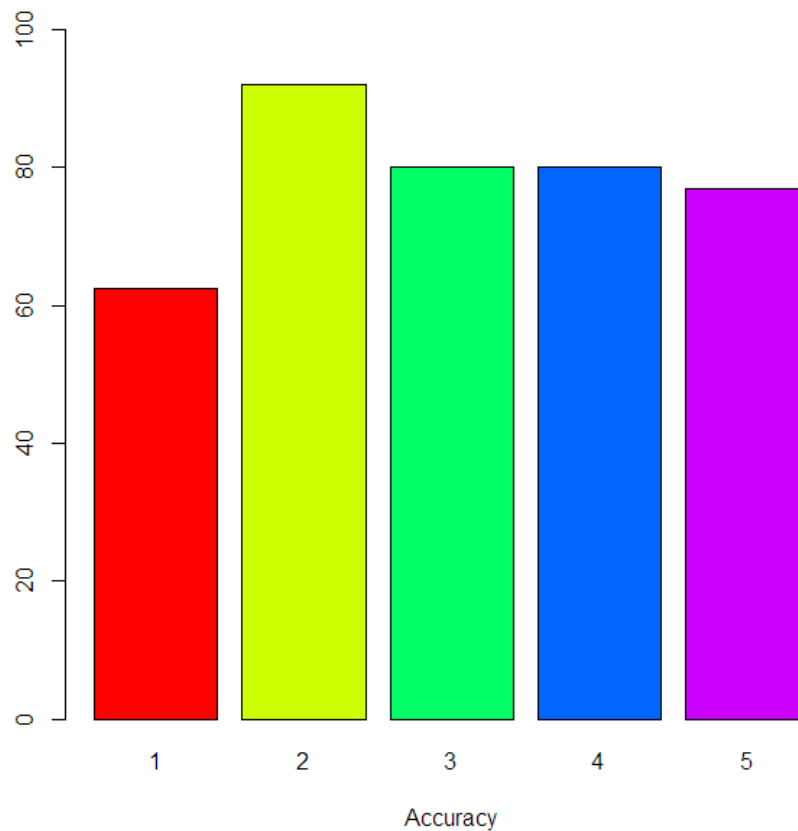


Figure 1. The percentage of accuracy from running naïve Bayes on different parts of data.

For the K-nearest neighbors algorithm we ran different kinds of methods first we chose three different K values: 1, 3 and 5 then we changed our measurement method from Euclidian to Manhattan which did not effect on our data. In Figure 2 to 4 you can see the percentage of accuracy we got from different values of K. We can see that there were no significant change from K=3 to K=5 and the results were almost the same but they have both better results than K=1 run.

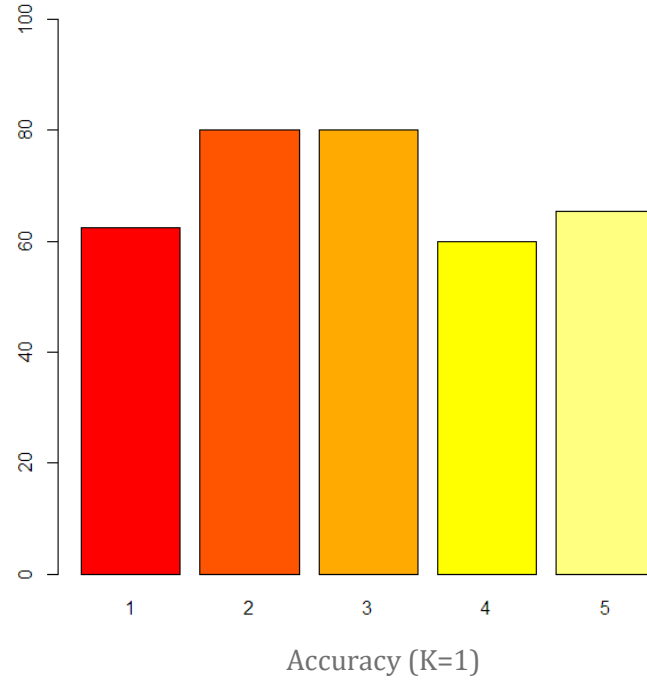


Figure 2 The percentage of accuracy from running K -nearest neighbors on different parts of data with the K value 1.

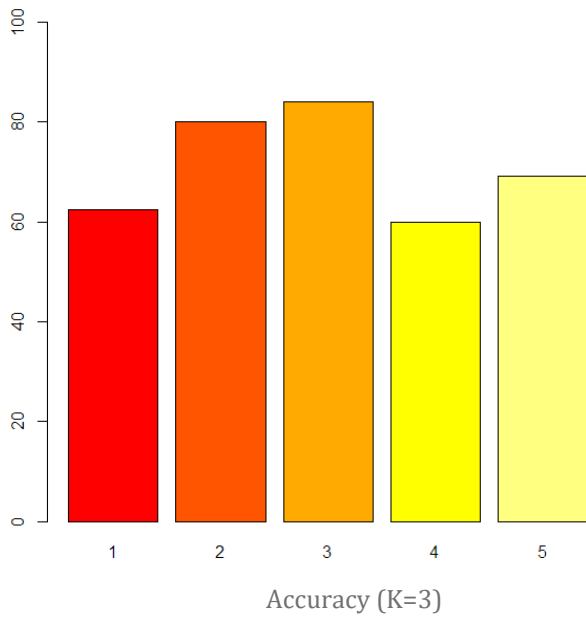


Figure 3 The percentage of accuracy from running K -nearest neighbors on different parts of data with the K value 3.

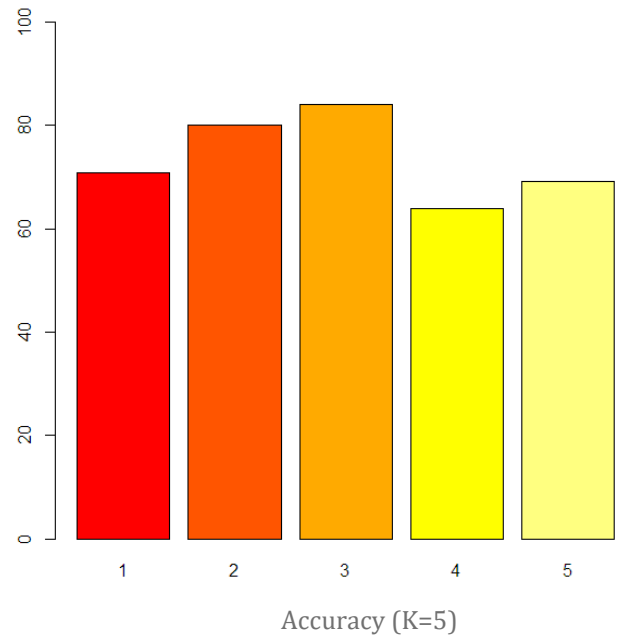


Figure 4 The percentage of accuracy from running K -nearest neighbors on different parts of data with the K value 5.

Overall in this problem we got better results in accuracy from naïve Bayes algorithm but it does not mean that naïve Bayes is a better algorithm.

The results we got from the multiple linear regression are shown in the plots below (Figure 5 to 8). Since this method is not a classification method it is not really meaningful to compare it with the other two methods.

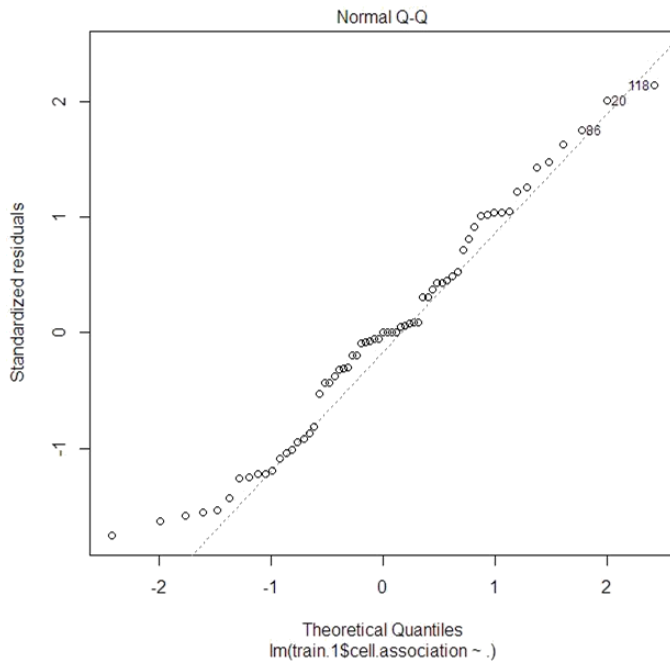


Figure 5 Theoretical Quantiles vs. Standardized residuals of multiple linear regression.

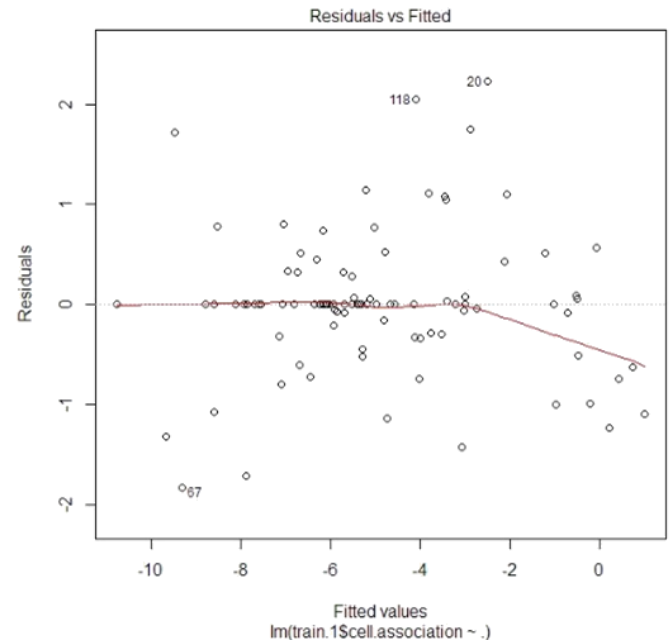


Figure 6 Fitted values vs. Residuals in multiple linear regression

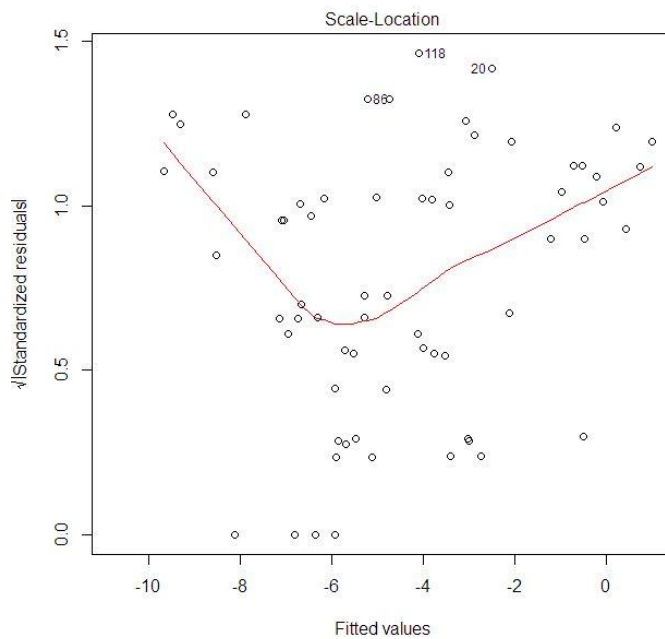


Figure 7 Scale Location of fitted values vs. radical of standardized residuals in multiple linear regression

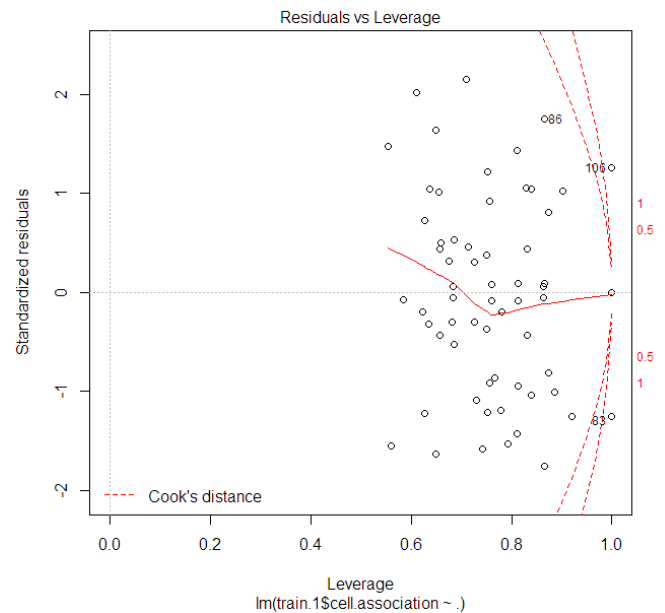


Figure 8 Standardized residuals vs. Leverage in multiple linear regression

DISCUSSIONS

The methods we used did not have any significant difference in results but naïve Bayes did give us a better answer even though there were relation between our features. In the future if we can reduce the effect of sparsity in our data which is a result of excessive number of features in comparison with the number of instances we could build much better models. The other thing we could work on is the number of different classes of toxicity in the dataset. At first we had three different classes of toxicity but two classes of these three were practically empty so we changed it to two classes. It did reduce our accuracy but the models were better and more meaningful.

In KNN method we saw that changing the K value had a direct influence on the accuracy so we could find a better K value for this method.

We also did use the mean of a column instead of missing values which is not a good estimation and we should work on that too.

To make these models better we have to have more understanding of the features we are using and use a dimension reduction method on them or find more instances to make our predictions more precise. The other thing we could work on is to get a better classification method to classify the toxicity we use in training set.

REFERNCES

1. Raies AB, Bajic VB. In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdisciplinary Reviews Computational Molecular Science*. 2016; 6(2):147-172. doi:10.1002/wcms.1240.
2. <http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf>
3. http://saedsayad.com/naive_bayesian.htm
4. www.saedsayad.com/k_nearest_neighbors.htm