Subject: ISYE6501

Assignment: HM1

Students (name, GT id):

×          Yuanting Fan, 904047984

×          Wenjia Hu, 904057780

×          Sen Yang, 904025995

# *Question 2.1*

**Describe a situation or problem from your job, everyday life, current events, etc., for which a classification model would be appropriate. List some (up to 5) predictors that you might use.**

**Scenario: The rapid return policy of China’s online shopping platform Taobao**

The Alibaba-owned e-commerce platform Taobao has introduced an eye-catching refund policy that allows consumers to receive quick refunds without even returning the products they bought. Although the policy initially aimed to improve the shopping experience for customers, it aroused widespread concerns among sellers as some consumers’ abuse of this “refund-only” policy may interrupt normal business operations and ultimately impact the benefits of merchants. Therefore, it is crucial for the platform to find a way to determine which buyers are eligible for refunds and which are not.

The platform can use a SVM classification model to categorize refund requests into two groups:

1) refund applications that are approved

2) refund applications that are rejected

Below are five predictors the platform can use in categorizing refund requests:

1. Product prices: refund applications for low-priced products are more likely to be approved. Extra caution should be exercised when handling refund requests for expensive merchandise.
2. Return rate of products: If a product has a higher rate of return rate, then it is more likely that it has defects and the buyer is more likely to receive refunds.
3. Seller’s tier on the platform: A seller’s tier on the platform reflects factors such as product quality, logistics, after-sales service. Therefore, it is less likely for buyers to receive refunds for products from a high-tier seller.
4. Buyer’s credit rating on the platform: A buyer’s good credit rating typically reflects a sound purchase history, making it more likely that their refund requests will be approved.
5. Buyer’s refund rate in the last 90 days: A high frequency of refund requests may suggest potential abuse of the return policy.

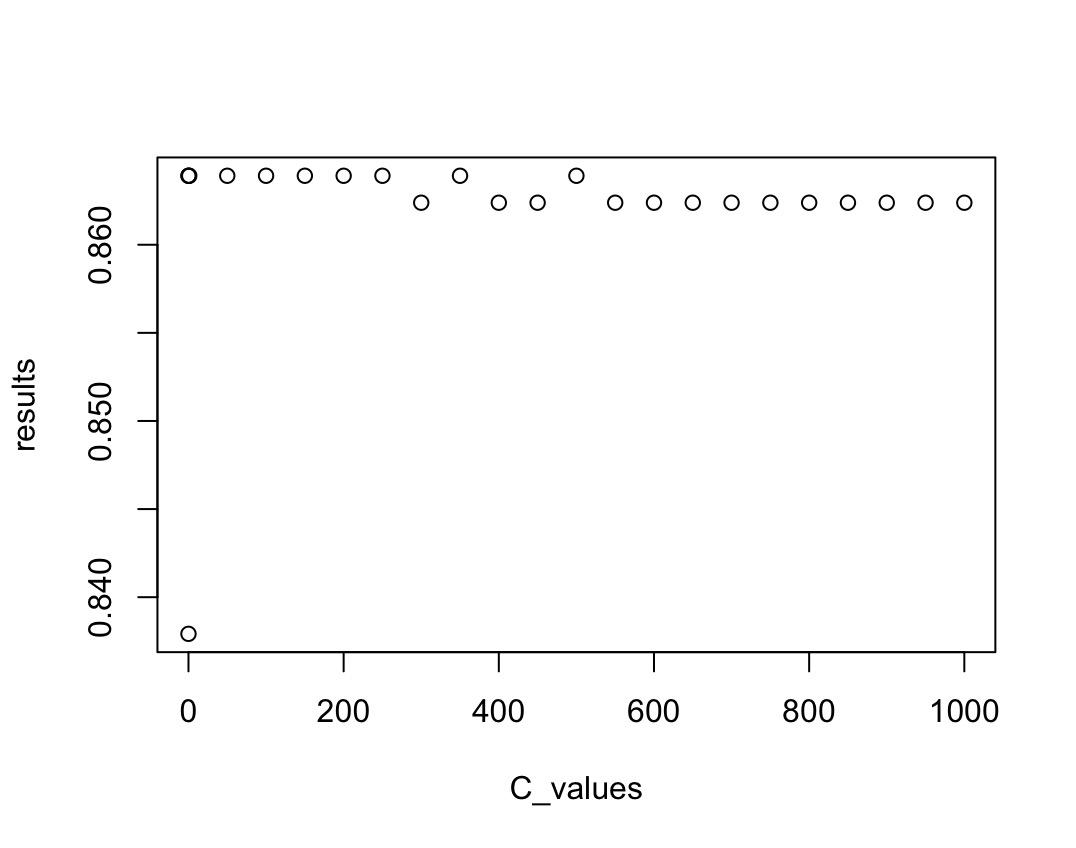
The SVM model is suitable for addressing this issue because:

1. The platform only needs to categorize all the refund requests into two groups. SVM works well in this regard.
2. Given the platform needs to deal with a large volume of refund data, SVM is an appropriate choice.

# *Question 2.2*

1. **Using the support vector machine function ksvm contained in the R package kernlab, find a good classifier for this data. Show the equation of your classifier, and how well it classifies the data points in the full data set. (Don’t worry about test/validation data yet; we’ll cover that topic soon.)**

We tried C values from 0 to 1000, step by 50, and plot C values with prediction accuracy:



It shows that when C varies from 0.001 to 1000, the correction rate does not change significantly. And the best performance of the model is 0.8639 (with C in a large range, for instance, either C=100 or 350, the prediction accuracy is 0.8639 in both cases).

In this case (takes C = 100 as an example, when the prediction accuracy is 0.8639), the equation is:

-0.0010065348 \* x1- 0.0011729048 \* x2-0.0016261967 \* x3 +0.0030064203 \* x4 +1.0049405641 \* x5 - 0.0028259432 \* x6 + 0.0002600295 \* x7 - 0.0005349551 \* x8 -0.0012283758 \* x9 + 0.1063633995 \* x10 + 0.08158492 = 0

In addition to the fundamental answer above, we also noticed several important results and will discuss them below:

1. **C values and the performance of the model**

We happened to find out that when C= 0.0014, the performance of the model is 0.8669, which is slightly better than the performance in our conclusion (0.8639). However, we decided to ignore that data, based on the following reasons:

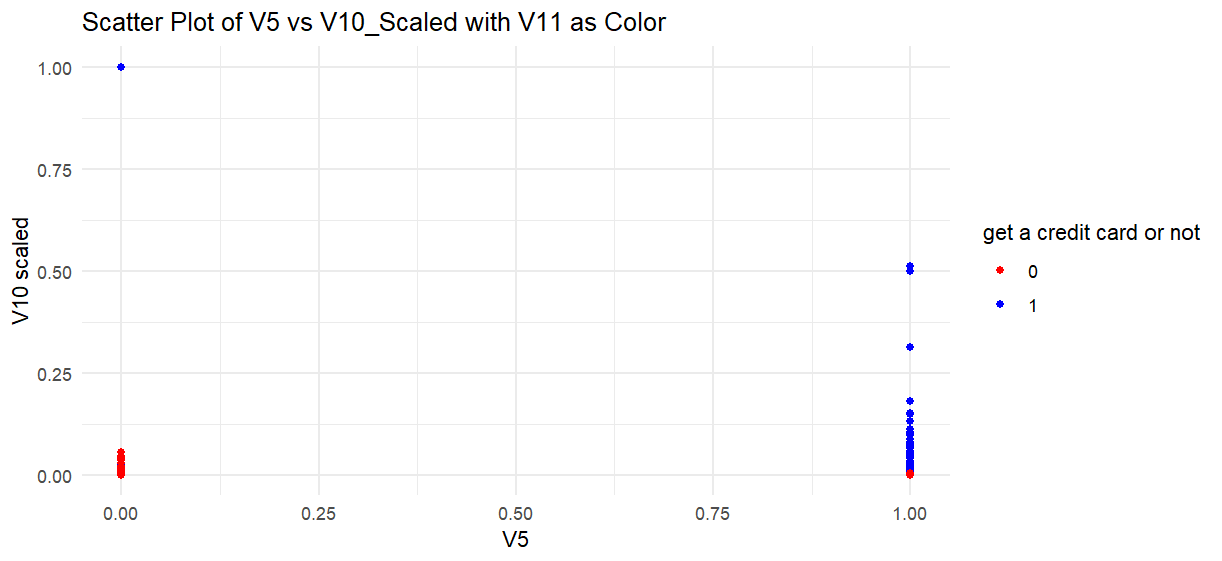
* When C is too small, almost all predictions will be “yes”
* The difference between 0.8669 and 0.8639 is not significant enough
* When considering the practical implications of the data set, we learned from its original source that this dataset is used to determine whether a bank should issue a credit card to an applicant. We believe that the consequences of incorrectly issuing a credit card could be severe. Therefore, our model predictions should be more conservative. Therefore, we should avoid choosing a C with a small value.
* We don’t have a validation set or a test set to determine whether the difference is due to a random effect. When C is too small, which means the SVM model is approximating a simple linear regression model because it only devotes to minimize the “error” without maximizing the “margin”. If the prediction accuracy is very good, it implies that the data set is probably simply linearly correlated, or it’s simply resulted from a random effect. We suppose that it’s a random effect since when C becomes larger, the prediction accuracy is also pretty good.

1. **Data variables effects:**

When C = 1000 (the results are similar regardless of C value):

a\_values= [0,0,0.0,1,0,0,0,0,0.11]

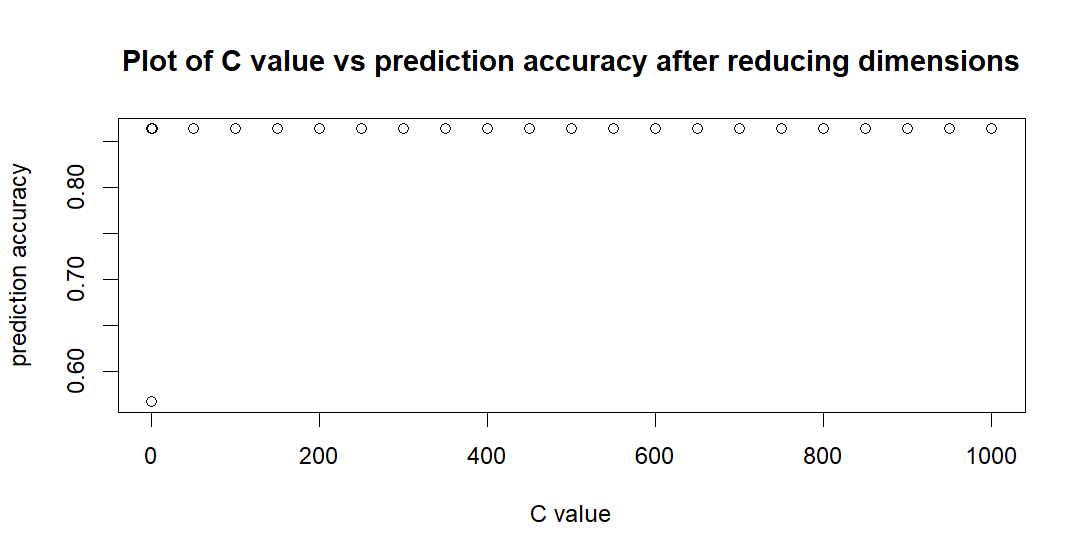
a0=-0.08

It means only V5 and V10 have essential effects on the V11, which represents whether the applicant was issued the credit card eventually or not. So, the data plotted only by V5 and V10 look like: 

V5 is categorical data with values of 0 and 1, while V10 is continuous ranging from 0 to 100,000 (probably means the salary). The chart seems to imply that if V5 equals 1 (if V10 is not zero), the applicant is likely to get the credit card application approval, unless V10 is big enough.

1. **Prediction accuracy after reducing dimensions:**

We then reduced the data set to contain only V5, V10 and V11: again, we tried C values from 0 to 1000, step by 50, and plot C values with prediction accuracy:

**The graph shows more reliability than before the reduction, it hugely decreases the prediction volatility resulted by the random effects, which attests to that indeed only V5, V10 are the effective predictors.**

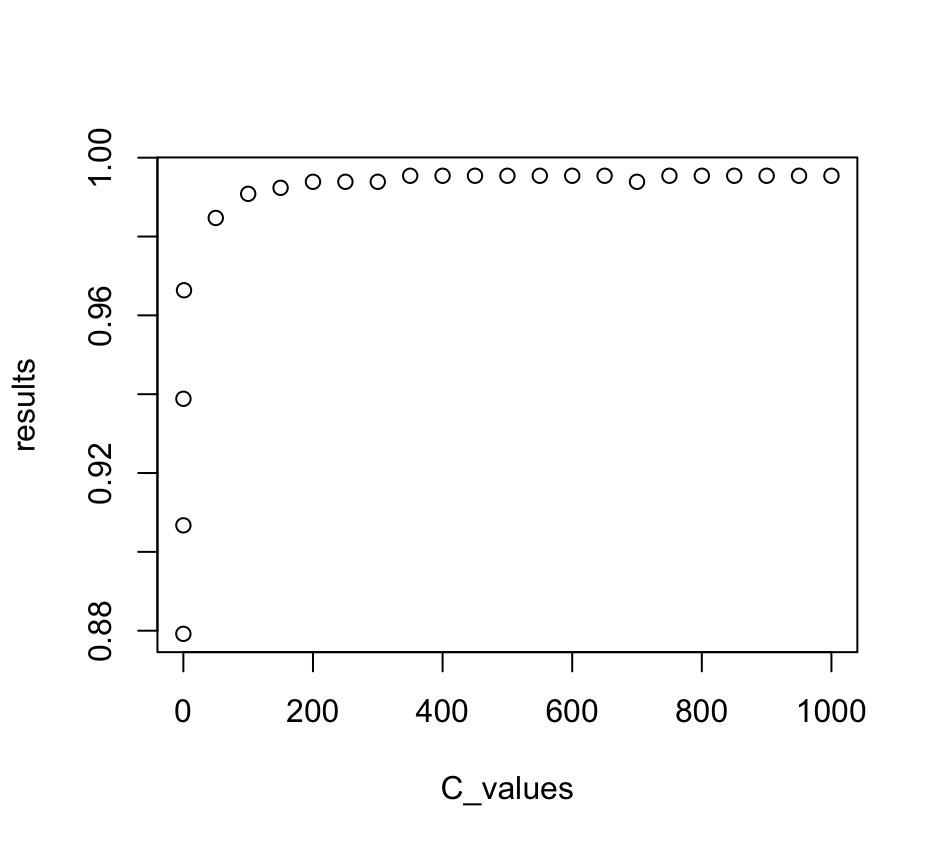
1. **You are welcome, but not required, to try other (nonlinear) kernels as well; we’re not covering them in this course, but they can sometimes be useful and might provide better predictions than vanilladot.**

We use a nonlinear model called polynomial(ploydot) in the ksvm library to compare the new model’s performance with the vanilladot model’s performance.

To do so, we take the following steps:

* Find out all the parameters that the polynomial model includes, and set the degree equals 3.
* Value the C from the same range and step as question 1, which is from 0 to 1000, step by 50 and calculate the prediction accuracy that every C makes.
* Find out the best C that enables the polynomial model to have the best performance: With the degree= 3, when C = 350, the polydot model’s prediction accuracy = 0.9954
* Compare the performance between the two models

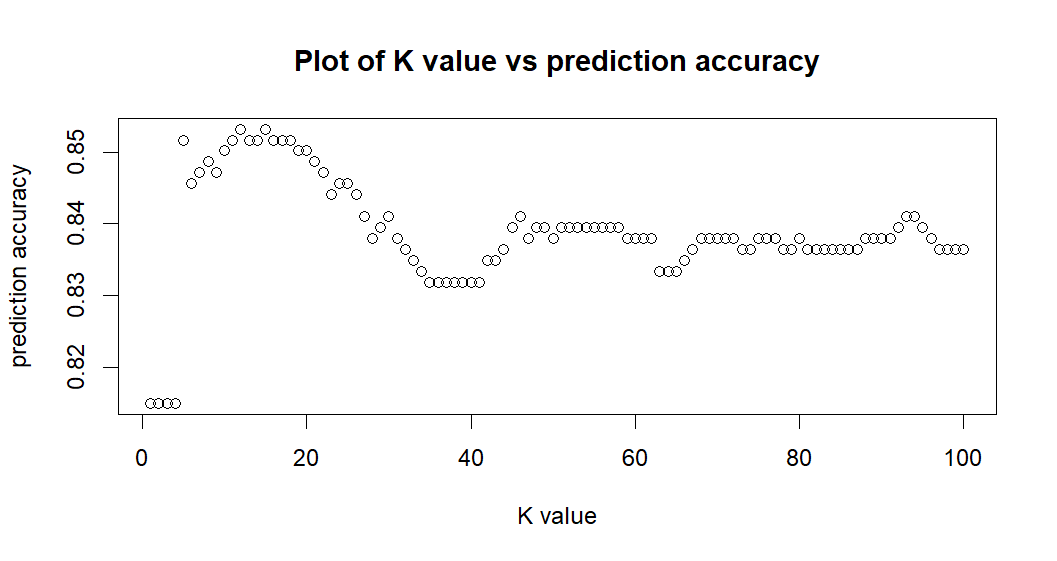
Results: the polynomial model provides a better prediction than the vanilladot model in this data set.



1. **Using the k-nearest-neighbors classification function kknn contained in the R kknn package, suggest a good value of k, and show how well it classifies that data points in the full data set. Don’t forget to scale the data (scale=TRUE in kknn).**

We use response for all but the ith data point as target response, predictors for all but the ith data point as training data, and ith data point as test data. In addition, we change the value of K to evaluate the prediction accuracy ratio. Thus, basically speaking, we run two loops: one is for each data point, and the other is for K value.

Here we obtain the plot relationship between K values and prediction accuracy:



After careful examination, we find that when K value = 12, the accuracy ratio goes up to 0.853211, which is the highest indicator.

Compared with the highest accuracy ratio of 0.8639 derived by SVM, KNN model is not as good.

**To summarize, non-linear SVM model seems more effective than linear SVM and KNN, although it probably requires more computing capacity.**