Aristoteles Cajiza - 1993274

Luke Campbell - 1569168

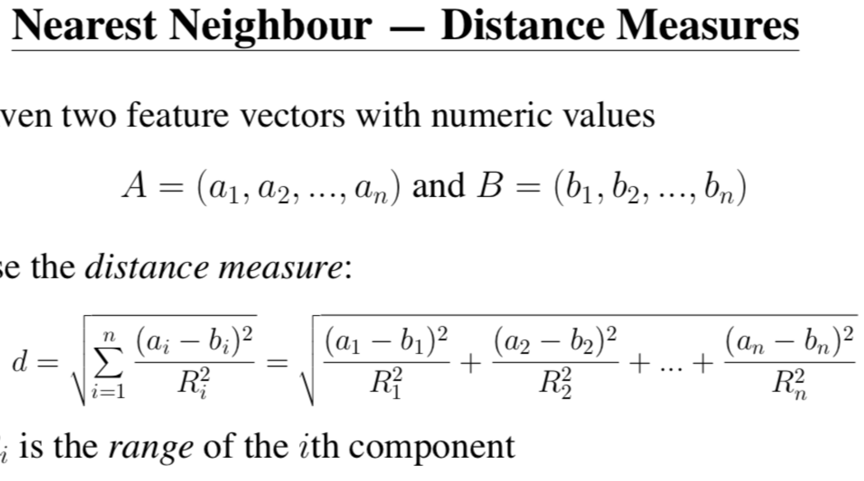
Dillon Zachariah - 1879187

Math 4323 Final Report

1. Introduction (Dillon Zachariah): This dataset contains information about the various types of posts that are uploaded to Facebook by its users in Thailand between the years of 2012-2018 . Namely that of Facebook pages that belong to 10 different Thai fashion and cosmetics retail sellers. Furthermore, the dataset used in this project was based upon social media interactions. More specifically the dataset aimed to gauge the effects of “facebook live” interactions versus normal posts. The data set contains one categorical variable “status_type” as well as the numbers of different interactions on posts. These interaction variables are number of reactions, number of comments, number of shares, number of likes, number of loves, number of wows, number of hahas, number of sads, and number of angrys. The response variable for this project is status_type. The goal of this project is to measure the different number of social media interactions based off of the status type to see if one status type gets greater interactions than the others.

2. Methodology (Aristoteles Cajiza): To analyze this dataset we used supervised learning to predict status_type. The methods of supervised learning to analyze the data are SVM and KNN. SVM is advantageous because it is relatively stable, it can handle linear and nonlinear data, and it can solve regression and classification. KNN is advantageous because it can be modified easily and efficiently, and it has no training period. KNN also uses “Feature Similarity”, this helps predict the value of any new data points. SVM has the disadvantages of having a long training time, and it is difficult to interpret. KNN has the disadvantages of being sensitive to overfitting, computationally expensive, and it does not work in higher dimensions like SVM. Supervised learning was chosen over unsupervised learning because we wanted to explore the relationships between the interactions and status_type.

   For KNN how we calculate the predictions for
   the observation based on its neighbors is given by:



Nearest Neighbour — Distance Measures

- Given two feature vectors with numeric values

$$A = (a_1, a_2, ..., a_n) \text{ and } B = (b_1, b_2, ..., b_n)$$

- Use the *distance measure*:

$$d = \sqrt{\sum_{i=1}^{n} \frac{(a_i - b_i)^2}{R_i^2}} = \sqrt{\frac{(a_1 - b_1)^2}{R_1^2} + \frac{(a_2 - b_2)^2}{R_2^2} + ... + \frac{(a_n - b_n)^2}{R_n^2}}$$

R_i is the *range* of the ith component

For SVM the hyper plane equations are:

Linear: $$K\left(x_i, x_{i'}\right) = \sum_{j=1}^{p} x_{ij}x_{i'j},$$

Polynomial: $$K\left(x_i, x_{i'}\right) = \left(1 + \sum_{j=1}^{p} x_{ij}x_{i'j}\right)^d$$

Radial: $$K\left(x_i, x_{i'}\right) = exp\left(-\gamma\sum_{j=1}^{p}\left(x_{ij} - x_{i'j}\right)^2\right)$$

3. Data Analysis(Luke Campbell, Dillon Zachariah): In the dataset there is one variable called status_published that was mainly represented by a bunch of ### symbols that we chose to omit because it did not translate well into R. When the data was being loaded into R four empty columns appeared and had to be removed. The variable status_id was not used as well because it was just an indexing variable. Since all the variables are counting the number of different interactions no scaling was needed.
    a. For SVM we will now train on 80% and test on 20% we will be using set.seed(2)
    train = sample(1:dim(facebookLikes)[1], dim(facebookLikes)[1] *0.8)
    testing = -train

Training a linear kinear:

```
tune.out1 = tune(svm, status_type~., data = facebookLikes[train,], kernel = "polynomial", ranges = list(cost=c(0.1, 1, 10), degree=c(0,1,2,3)))

summary(tune.out1)
```

This gave us an output of:

```
Parameter tuning of 'svm':


- sampling method: 10-fold cross validation


- best parameters:
 cost
   10


- best performance: 0.2468085
- Detailed performance results:
  cost     error dispersion
1  0.1 0.2471631 0.01621153
2  1.0 0.2469858 0.01527638
3 10.0 0.2468085 0.01549771
```

```
svmfit1 = svm(status_type~., data = facebookLikes[train,], kernel = "polynomial", degree = 1, cost = 10)
```

Training for a polynomial model:

```
tune.out1 = tune(svm, status_type~., data = facebookLikes[train,], kernel = "polynomial", ranges = list(cost=c(0.1, 1, 10), degree=c(0,1,2,3)))
summary(tune.out1)
```

This gave us the output of:

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost degree
   10      1

- best performance: 0.2475177

- Detailed performance results:
  cost degree     error dispersion
1  0.1      1 0.2524823 0.01751644
2  1.0      1 0.2478723 0.01565469
3 10.0      1 0.2475177 0.01438491
4  0.1      2 0.3035461 0.02109192
5  1.0      2 0.2799645 0.01630285
```

```
6 10.0      2 0.2682624 0.01796632
7  0.1      3 0.3109929 0.02203787
8  1.0      3 0.2806738 0.01622984
9 10.0      3 0.2549645 0.01475886
```

```
svmfit1 = svm(status_type~., data = facebookLikes[train,], kernel = "polynomial", degree = 1, cost = 10)
```

Training for a radial model:

```
tune.out2 = tune(svm, status_type~., data = facebookLikes[train,], kernel = "radial", ranges = list(cost=c(0.1, 1, 10), gamma=c(0.5, 1, 2)))

summary(tune.out2)
```

This gave us the output:

```
Parameter tuning of 'svm':


- sampling method: 10-fold cross validation


- best parameters:
 cost gamma
   10     2


- best performance: 0.1368794


- Detailed performance results:
  cost gamma     error  dispersion
1  0.1   0.5 0.2476950 0.013030541
2  1.0   0.5 0.2200355 0.014511833
3 10.0   0.5 0.1625887 0.009195959
4  0.1   1.0 0.2478723 0.014993660
5  1.0   1.0 0.1710993 0.009428520
6 10.0   1.0 0.1404255 0.008223417
7  0.1   2.0 0.2039007 0.009382097
```

```
8  1.0   2.0 0.1464539 0.008112227
9 10.0   2.0 0.1368794 0.006779967
svmfit2 = svm(status_type~., data = facebookLikes[train,], kernel = "radial", gamma = 0.5, cost = 1)
```

b. For KNN we do the same

```
x.train <- facebookLikes[train, -2]
x.test <- facebookLikes[testing, -2]

y.train <- facebookLikes$status_type[train]
y.test <- facebookLikes$status_type[-train]
knn.pred <- knn(train = x.train, test = x.test, cl = y.train, k = 1)
mean(knn.pred != y.test)
[1] 0.5120567
knn.pred <- knn(train = x.train, test = x.test, cl = y.train, k = 3)
mean(knn.pred != y.test)
[1] 0.4978723
knn.pred <- knn(train = x.train, test = x.test, cl = y.train, k = 7)
mean(knn.pred != y.test)
[1] 0.4673759
knn.pred <- knn(train = x.train, test = x.test, cl = y.train, k = 10)
mean(knn.pred != y.test)
[1] 0.4411348
```

c. Between both models the best performance achieved was KNN when k = 1 had an error percent of 51.2% with the second best being 49% when k = 3

d. We will fit the model with no training/test subdivide on the KNN when k = 3

```
mean(knn.pred != facebookLikes$status_type)
[1] 0.08695035
summary(knn.pred)
  link  photo status  video
    33   4549    262   2206
> summary(status_type)
  link  photo status  video
    63   4288    365   2334
```

e. When doing testing and training we got a better performance than on the unsplit dataset. When testing on a KNN model on the training and testing we got performance of about 50% which is decent considering the size of the data and KNN is better for small data. This is most likely because the status_type is related to all the other variables so it acts more like a graph with nodes than an SVM would.

4. Conclusion (Aristoteles Cajiza, Dillon Zachariah, Luke Campbell): Based on the KNN model unsplit I would say that the model predicted much higher interactions with photos and lower with the other 3 types (link, status, video). We can conclude that this means people like interacting with links, status updates, and video than predicted. The "facebook live" interactions have caused people to interact with status types that feel more alive than a photo than predicted. We found this data interesting because it's data from a social platform that is used by millions of people daily and has a diverse user base from around the world. Although this data is limited to posts from Thailand, the interactions with the posts can be from virtually anyone. Furthermore, it allows us to gain some insight as to how successful businesses can be through social media alone, and how much potential customers will engage with these posts depending on the format. It was also very interesting to delve deeper into the data and see how exactly the introduction of live streaming to the platform influenced user engagement on these seller's pages. We would like to try unsupervised learning methods at some point and see how the results would vary compared to supervised learning.

5. References: Dataset

https://archive.ics.uci.edu/ml/datasets/Facebook+Live+Sellers+in+Thailand#
Model formula: https://stackoverflow.com