Final Project: Machine Learning Scenario Report

Jonathan Rissew

Southern New Hampshire University

IT460: Machine Learning

Joshua Montgomery

October 15, 2023

Identification and Defense

This project will use the "User Identification from Walking Activity" dataset provided by UC Irvine. A model will be constructed using a support vector machine (SVM) to classify and predict users identities based on accelerometer data. When the model is complete it can be further extended for use in user identification based on motion sensor data from cell phones. The SVM model was selected due to its dual purpose of application in classification and regression (Lantz 2019, p. 23). A similar data set using an SVM on accelerometer data was able to identify human postures and transitions to other postures with high degrees of certainty (Rodriguez-Martin 2013). This helps to solidify the choice to use an SVM model, as it has been used for similar applications with success in the past.

As mentioned previously, SVM was selected over other models due to its ability to classify and predict. A regression model may do well with forecasting and time-series data, but it does not handle classification. The process of identifying user's based on sensor data requires the model to classify data into categories. Regression models work best when predicting a continuous variable. Other classification models are also valid for modeling this dataset. Other classification algorithms like k-nearest neighbor, naive bayes and k-means clustering could be used with great success. SVM was selected over them due to its application in similar data sets in the past, as referenced above, and due to its handling noise very well (Lantz, p. 248). K-means clustering would likely be a great option as well, as it is an unsupervised method for classifying the data.

The ability to identify users based on motion data is one that should be treated cautiously. In many ways it could invade one's privacy. Yet, there is genuine benefit from being able to derive this information. The study performed by Rodriguez-Martin was able to use motion sensor data to identify various postures of the users in their data set. This was applied with great accuracy in predicting the posture of people with Parkinson's disease (2013). In the case of that study, the ability to identify the posture of a patient can be useful in providing medical support in predicting the user suddenly falling or becoming unresponsive. The data set provided in this project is not as robust as the one in that study and will only be able to, at best, predict users based on their walking down a path. This still provides a use case in the real world. For example, if a criminal has stolen someone's phone, the phone manufacturer may be able to assist law enforcement to ascertain the criminal's identity using accelerometer data.

SVM will be applied using the kernlab package in R. The data will be trained using half the data set per user. The trained model will then be tested for its predictive ability against the remaining half of the data set. The model will undergo optimization to see if predictive ability can be increased. This will be done through experimenting with various kernels to see if accuracy improves. The model can then be tested for an optimum cost parameter to land on a final optimized model (Lantz 2019, p. 255-259).

SVM is a strong tool in machine learning due to its dual purpose use case and its ability to manage noise. However, it has drawbacks in that it takes time to optimize the model and the model can be interpreted (Lantz 2019, p. 248). Due to its use in similar cases as shown in the study by Rodriguez-Martin et al, the SVM is a solid choice in the provided use case.

Model Execution

The data must be organized to properly train the model. As it was provided, the data is split into 22 files for each participant in the collection process. There are also no header names for the data in the file. Each file is loaded into an R environment where headers are attached and a new column is filled with the user number. Once all files have been organized they are merged into a single walk_data dataframe.

```
person4 <- read.csv("4.csv", header=FALSE)
colnames(person4) <- c("time","xaccel","yaccel","zaccel")
person4$num <- 4
```

```
> walk_data <- rbind(person1, person2, person3, person4, person5, person6, person7, person8, person9, person10, person11, per
son12, person13, person14, person15, person16, person17, person18, person19, person20, person21, person22)
```

The amount of observations per person can be found using the table function. It can be seen that some people have significantly more observations than others. Participant 19 has only 911, the lowest in the group. In contrast, several have over 20000 observations.

```
> table(walk_data$num)

   1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20 
5069  3882  1144  6981  1129  4936  3729  3457  7988  3086  5636  4799  6699 12027  3653  1728 21991 20758   911 16949 
  21    22 
3082  9698 
```

The data frame was now randomized before building out the model. The randomized model is stored in walk_data_rand. A comparison of the data after and before randomization is shown using the str function.

```
> RNGversion("3.5.2")

> set.seed(123)
> walk_data_rand <- walk_data[sample(nrow(walk_data)),]

> str(walk_data_rand)
'data.frame':	149332 obs. of  5 variables:
 $ time  : num  48 624.4 79.7 370 25.4 ...
 $ xaccel: num  -3.678 -4.59 -1.417 -1.689 0.844 ...
 $ yaccel: num  9 5.75 6.28 10.38 9.7 ...
 $ zaccel: num  0.0409 -1.6481 2.4517 -0.8445 -2.833 ...
 $ num   : num  11 18 14 20 22 2 17 20 17 14 ...
> str(walk_data)
'data.frame':	149332 obs. of  5 variables:
 $ time  : num  0 0.0306 0.0698 0.0998 0.1298 ...
 $ xaccel: num  0.695 0.15 -0.3 -1.689 -2.179 ...
 $ yaccel: num  3.174 3.487 1.948 1.417 0.953 ...
 $ zaccel: num  7.5 9.28 9.11 10.12 10.92 ...
 $ num   : num  1 1 1 1 1 1 1 1 1 1 ...
```

The summary function shows that all accel fields have similar maximum values around 19.3-19.5 but varied minimum values ranging from -19.5 to -10.9. The kernlab implementation will normalize values by default so no further modification is needed.

```
> summary(walk_data_rand$xaccel)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-19.57200  -3.14630  -1.30760  -1.65533  -0.04086  19.31400
> summary(walk_data_rand$yaccel)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-10.924   7.164   8.853   8.769  10.338  19.572
> summary(walk_data_rand$zaccel)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-14.98200  -1.22580  -0.08172   0.55558   1.53910  19.34100
```

The num column, which denotes the person's numerical ID was set as a factor for modeling purposes. Two new data frames are constructed for training and testing. The training model has 120,000 observations and the testing model has the remaining 29,332 observations.

```
walk_data_rand$num <- as.factor(walk_data_rand$num)

walk_train <- walk_data_rand[1:120000,]
walk_test <- walk_data_rand[120001:149332,]
```

The first SVM model will be using the vanilladot kernel.

```
> walk_classifier <- ksvm(num ~ ., data = walk_train, kernel = "vanilladot")
 Setting default kernel parameters
> walk_classifier
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 115652
```

The model was then tested against the testing subset and the results printed.

```
walk_predictions <- predict(walk_classifier, walk_test)
```

```
> table(walk_predictions, walk_test$num)

walk_predictions    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22
              1   250    1    0  112    8    0    0   55    9   80   93  106    6    1   11   25   66   75    0    0    1   89
              2     3    7    0    2    0    0    0    1   21    2    2    2    0    0    8    4    6    3    0    9    5    3
              3     1    3   34    0    1   34   17    1    4    0    6   21   15    1   22   20    4   20    0    4    8    2
              4    73   20    0  198    0    1    0  130    2  106  183    1    3   28   15    9   95  159    0    0    1   57
              5    17    6   10    3  117    1    7    1    3    4    5    4    5   21    1   19   44   39   42   27    4    1
              6     9   83   56    4    0  234   89    1   26    0    4  164   94   30   34   11    3    6    0  102  108    6
              7    15   29   40   13    7  106  469   12   15    0   16   51   70   11   19   13    7   50    0   32    3    1
              8    11   18    0   30    0    3    2   20    1   14   41    0    5   24   16   21   29   16    0    0    1   22
              9     2   18    0    5    0    4    0    4   16    6    3    3   10    2    8    0    2    6    0   13    3    4
              10  130   16    0  101    7    2   13  157    5  147  133    2    5   24   32   34   77   45    0    2    8   97
              11   20    0    0   36    0    2    0   64    1   28   52    0    5    3   19   20    7   45    0    0    0   55
              12   74   22   17   73    1   40   21   60   49   42   52   52   62   22  130   34   41   77    0   38   33   60
              13    4   99    0   17    1  261   20    4   10    0    4   86  314   81   29    2   18   21    0   12  189    6
              14   53   69   12  177   49   31   59   43   90   15   72   51  204 1655  115   44  228  457    5   26   34   88
              15    7   88    1   16    0   11    0   17   24   22   21    3    7   12   77   20   49   17    0   21   39   27
              16    5   16    0    6    1    6    1    2    4    0    7    8   27   10    3    5   23    8    0    1   17    9
              17  142    0    0  381    0    0    0    7   28    3  190  195   62  333    0    0 3122 2235    0  927    0  634
              18    3    1    0   16    0    1    0    0    9    0    9    0   36   57    0    0   18   41    0   42    0   50
              19    5    5    9   13   25    9   11    8   13   13    3   18   39   22    6   21   14   20  135    5    4   21
              20   60  140   10   36    1  167   66   14 1051   69   58   90  239   39   83   23  427  572    0 1905   60  318
              21    5   58   23    1    0   61    0    0   93    5    4   68   33   22   32   34    2   24    0   59  108    4
              22  127   73    0  145    0    2    0   89   26   36  161    0   47   25   67   12   61  107    0   51   11  308
```

As can be seen, the model's accuracy does not appear to be high. Using prop.table we can see that the model was able to predict the correct user 31.6% of the time. This is a significant result as random guessing would be a 1/22 or 4.5% chance of a correct prediction. However, 31.6% is still too low to provide much utility.

```
> agreement <- walk_predictions == walk_test$num
> prop.table(table(agreement))
agreement
    FALSE      TRUE 
0.6840993 0.3159007 
```

To attempt to improve model accuracy another kernel will be tested. Many recommend using Gaussian RBF as a starting point so this kernel will be selected for this test. When checking the results it can be seen that the model has risen to a 54.7% success rate in predictive ability.

```
walk_classifier_rbf = ksvm(num ~ ., data = walk_train, kernel = "rbfdot")
walk_predictions_rbf <- predict(walk_classifier_rbf, walk_test)
```

```
agreement_rbf <- walk_predictions_rbf == walk_test$num
```

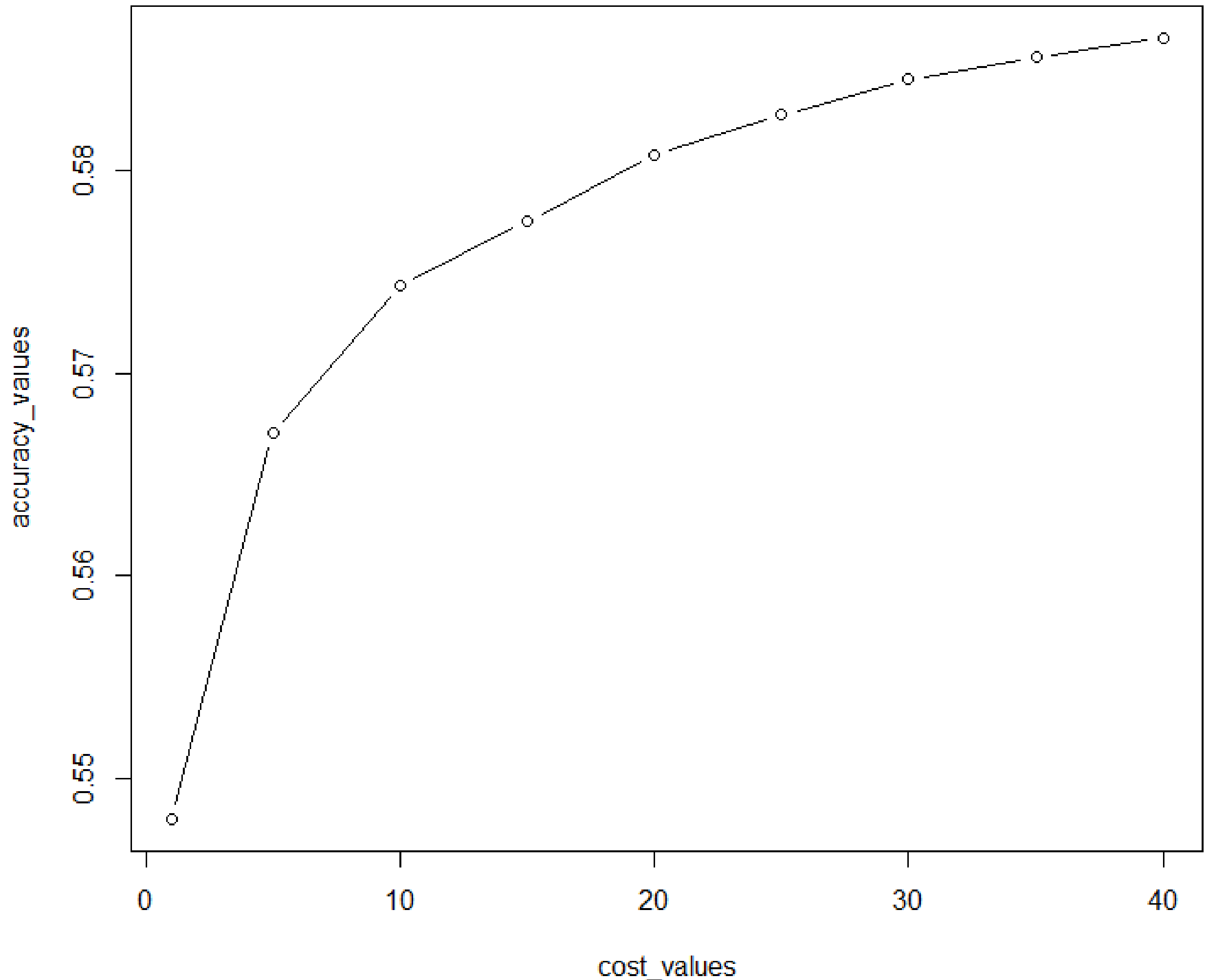
walk_predictions_rbf	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	431	3	0	117	4	5	9	45	5	45	92	9	3	0	12	9	59	47	0	1	1	99
2	8	101	1	13	0	21	2	32	46	16	31	15	18	22	57	6	42	16	0	33	51	36
3	1	1	65	0	0	37	21	0	7	0	3	27	14	2	24	14	2	8	0	10	9	4
4	83	5	0	526	3	0	3	37	12	23	81	4	15	10	16	22	87	125	0	3	0	41
5	7	5	12	2	181	4	0	3	7	2	4	0	13	12	4	15	16	6	3	6	4	6
6	2	71	26	6	0	403	41	2	14	0	3	84	110	19	18	6	2	6	0	45	57	8
7	7	33	27	0	0	84	512	9	13	0	12	59	50	11	18	13	3	1	0	24	8	1
8	17	15	0	41	0	2	4	156	5	29	85	0	9	14	22	25	66	58	0	0	1	80
9	26	111	4	60	1	72	12	12	838	52	48	21	87	5	55	35	80	81	0	412	61	140
10	62	13	0	52	1	2	2	84	19	208	80	0	1	1	11	33	32	52	0	14	9	51
11	78	13	0	97	0	4	0	121	8	105	350	1	6	3	21	5	103	103	0	0	3	71
12	7	14	28	7	0	62	28	4	18	0	1	404	40	3	18	5	12	8	0	11	25	1
13	16	52	6	25	4	101	34	4	12	0	20	50	502	57	47	9	23	40	0	30	22	22
14	19	35	0	33	7	12	11	24	8	12	13	20	126	2048	28	37	65	83	0	12	7	25
15	10	52	0	18	0	10	2	18	12	7	10	10	27	18	191	30	20	36	0	4	11	30
16	1	1	0	7	1	6	6	4	3	1	4	0	0	0	4	17	5	1	0	4	0	1
17	59	15	22	133	4	2	0	10	20	2	65	21	8	37	12	33	3101	314	0	290	11	56
18	21	18	0	55	2	12	24	36	8	17	9	36	38	56	45	7	248	2527	0	113	23	126
19	0	0	0	6	9	4	4	3	2	2	1	0	23	4	3	6	1	8	179	0	0	2
20	33	98	12	20	1	69	56	6	384	17	25	100	105	36	57	14	241	267	0	2182	41	203
21	5	61	9	2	0	62	4	0	26	1	5	63	32	20	14	9	13	15	0	28	286	10
22	123	55	0	165	0	2	0	80	33	53	177	1	61	45	50	21	122	241	0	54	7	849

```
> prop.table(table(agreement_rbf))
agreement_rbf
    FALSE      TRUE
0.4525774 0.5474226
```

Another piece of the model that can be tuned is the cost parameter. We will use some automated methods to test the cost parameter to see if the accuracy of the model can be increased.

```
cost_values <- c(1, seq(from = 5, to = 40, by = 5))

> accuracy_values <- sapply(cost_values, function(x) {
+     set.seed(12345)
+     m <- ksvm(num ~ ., data = walk_train, kernel = "rbfdot", C = x)
+     pred <- predict(m, walk_test)
+     agree <- ifelse(pred == walk_test$num, 1, 0)
+     accuracy <- sum(agree) / nrow(walk_test)
+     return(accuracy)
+ })
> plot(cost_values, accuracy_values, type = "b")
```



Increasing the cost parameter to 40 results in a model with 58.6% accuracy, almost 5% improvement. This can be seen further with the new table of results.

```
> walk_classifier_rbf40 = ksvm(num ~ ., data = walk_train, kernel = "rbfdot", C=40)
> walk_predictions_rbf40 <- predict(walk_classifier_rbf40, walk_test)
> agreement_rbf40 <- walk_predictions_rbf40 == walk_test$num
> prop.table(table(agreement_rbf40))
agreement_rbf40
    FALSE      TRUE
0.4137461 0.5862539
> table(walk_predictions_rbf40, walk_test$num)

walk_predictions_rbf40    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22
                    1   508    1    0  101    3    7    3   38    4   30   80    8    2    1   10    7   64   45    0    2    3   77
                    2    13  171    1   10    3   28    7   14   46   23   29   28   29   23   38   12   50   21    0   47   61   27
                    3     4    1  108    0    1   36   20    2   10    0    3   34   22    4   23   16    2   12    0   10   10    4
                    4    77    4    0  607    7    0    2   32   14   25   78    3   12   12    8   24   59   99    0    1    0   52
                    5     4    0    6    2  186    1    0    0    2    1    1    0   18   10    4    1    5    7    1    1    1    0
                    6     7   54    7   11    0  434   34    3   14    1    4   52  108   10   22   10    3    6    0   43   46    5
                    7     3   30   21    1    0   56  541    8   13    0    9   59   46    9   16    8    5    1    0   25    9    2
                    8    28    9    0   47    0    2    5  198    4   33   55    0    9   16   22   24   60   62    0    2    5   77
                    9    20   70    1   37    1   66   14    9  898   39   29   27   71   10   39   30   61   74    0  346   55  115
                    10   55   12    1   58    1    3    2  104   24  253   99   16    8    1   12   33   53   51    0   17   19   64
                    11   67   13    0   90    0    0    0   83    4   77  364    1    8    3   16    6   82   88    0    1    2   56
                    12    9   26   24    5    0   97   30    8   16    3    2  464   46    5   20    5   14    8    0   18   35    4
                    13   10   42    4   29    2   70   30    1   13    6   19   41  573   50   26   12   17   44    0   33   21   32
                    14   17   25    0   22    5   11    9   19    6    4   10   13   82 2071   19   28   50   73    0   13    6   15
                    15   12   47    1   20    1   17   11   28   14    7   15   10   32   31  275   27   17   33    0    4   10   37
                    16    2    0    1    8    0    3    5    3    2    0    3    0    0    0    5   29    3    1    0    1    0    2
                    17   48   31   22  102    2    6    0   14   20    3   90   17    7   34   18   34 3233  280    0  293   10   46
                    18   17   15    0   44    2    5    0   33   14    8   20   18   30   37   20   12  222 2623    2   90    0  101
                    19    0    0    0    2    4    2    3    2    1    1    2    0    4    1    2    4    1    4  179    0    0    0
                    20   39  104    8   18    0   79   50    6  322   11   34   90   99   31   57   14  204  271    0 2246   49  192
                    21    2   58    6    2    0   51    4    5   22    1    4   38   26   16   15    8   10    9    0   23  290    9
                    22   74   59    1  169    0    2    5   80   37   66  169    6   56   48   60   27  128  231    0   60    5  945
```

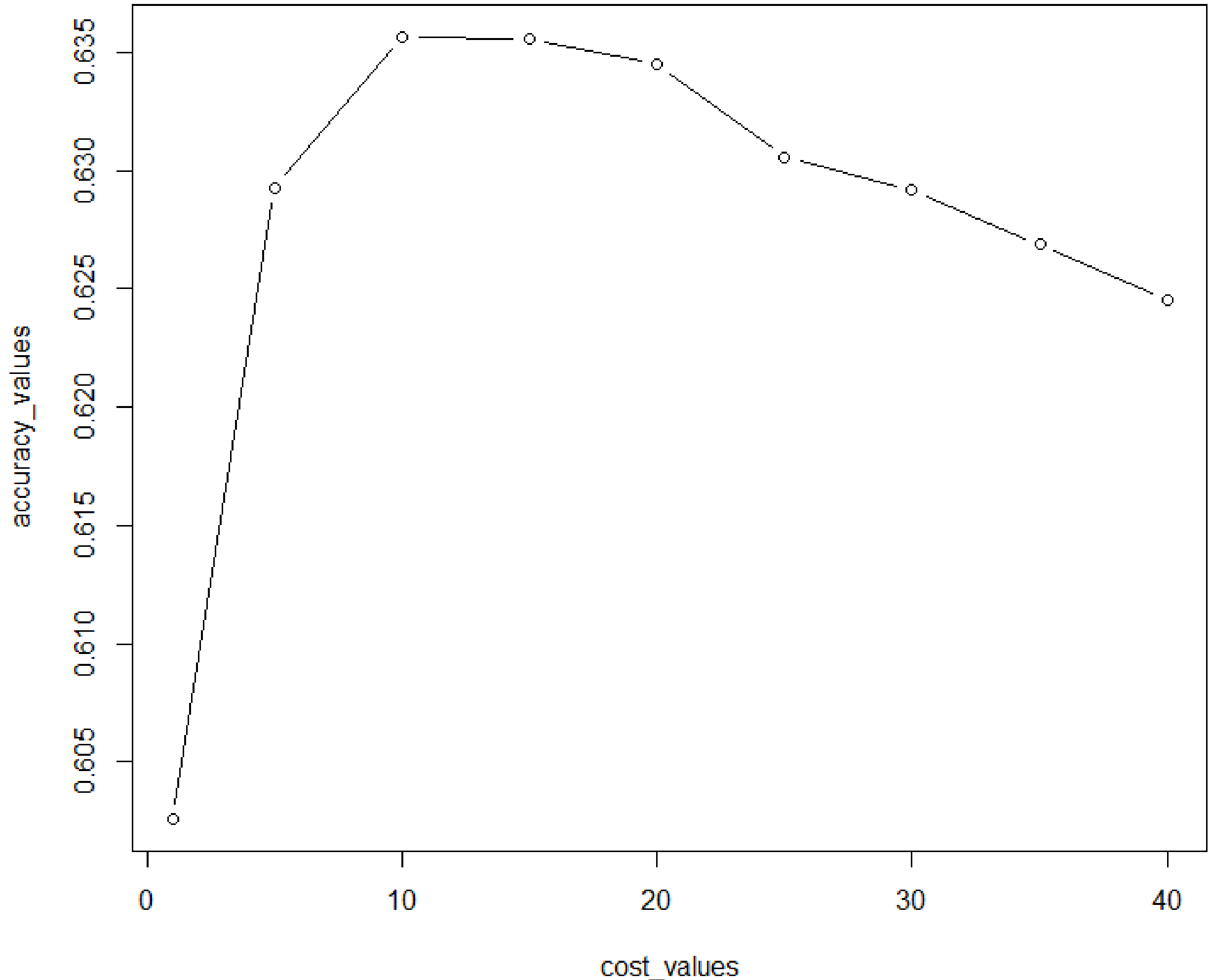
.

The model will be tested with one more kernel, the laplace kernel. The laplace kernel without cost parameter optimization outperforms the optimized RBF kernel model with a 60.3% accuracy.

```
walk_classifier_laplace = ksvm(num ~ ., data = walk_train, kernel = "laplacedot")
walk_predictions_laplace <- predict(walk_classifier_laplace, walk_test)
agreement_laplace <- walk_predictions_laplace == walk_test$num
```

```
> prop.table(table(agreement_laplace))
agreement_laplace
    FALSE      TRUE
0.3974499 0.6025501
```

```
> accuracy_values <- sapply(cost_values, function(x) {
+     set.seed(12345)
+     m <- ksvm(num ~ ., data = walk_train, kernel = "laplacedot", C = x)
+     pred <- predict(m, walk_test)
+     agree <- ifelse(pred == walk_test$num, 1, 0)
+     accuracy <- sum(agree) / nrow(walk_test)
+     return(accuracy)
+ })
> plot(cost_values, accuracy_values, type = "b")
```




```
> walk_classifier_laplace10 = ksvm(num ~ ., data = walk_train, kernel = "laplacedot", C=10)
> walk_predictions_laplace10 <- predict(walk_classifier_laplace10, walk_test)
> agreement = walk_predictions_laplace10 == walk_test$num
> prop.table(table(agreement))
agreement
    FALSE      TRUE 
0.3644143 0.6355857 
```

Cost optimization on the laplace kernel SVM model shows that a cost of 10 produces the highest accuracy of 63.6%. Thus our final model is an SVM model using the laplace kernel with a cost value of 10.

```
> table(walk_predictions_laplace10, walk_test$num)
```

walk_predictions_laplace10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	534	3	1	104	5	13	1	31	11	24	64	11	5	2	10	9	51	40	0	12	3	45
2	7	296	2	8	1	12	5	7	20	9	17	14	35	25	37	7	18	18	0	26	35	22
3	4	1	137	0	0	28	17	1	10	0	2	25	12	4	17	15	0	2	0	4	13	3
4	63	5	0	650	6	0	2	33	4	27	56	3	13	12	6	30	43	51	0	1	2	48
5	4	0	3	1	188	1	0	3	1	1	1	0	10	6	2	3	4	3	3	4	1	0
6	9	54	10	8	1	481	55	2	19	1	7	70	102	10	30	9	3	3	0	52	51	11
7	6	18	16	0	0	59	523	8	11	0	10	42	35	7	15	8	4	1	0	27	7	2
8	22	11	0	44	0	0	5	220	2	19	42	0	6	15	16	13	30	30	0	1	5	34
9	17	58	3	18	0	45	17	9	945	19	19	26	42	6	40	20	28	67	0	254	36	76
10	46	5	1	54	1	4	2	89	20	278	82	6	9	1	12	25	26	41	0	7	9	69
11	84	17	0	95	1	6	1	81	10	63	519	0	12	2	22	16	90	81	0	16	15	88
12	6	33	11	10	0	77	32	4	23	7	5	534	59	17	19	6	15	7	0	27	37	8
13	8	42	9	28	1	75	48	4	20	3	15	47	679	48	20	16	21	34	0	38	29	30
14	20	20	2	14	4	11	7	18	6	1	7	7	67	2083	15	28	43	52	0	15	6	10
15	9	41	1	27	3	15	8	24	19	8	19	12	23	23	329	25	11	36	0	12	12	40
16	1	2	1	12	0	3	5	8	7	10	3	2	5	11	9	67	5	4	0	3	1	5
17	37	21	2	91	1	5	6	24	28	12	63	11	11	27	16	15	3397	259	0	194	8	67
18	31	19	0	73	3	1	2	49	16	24	35	14	22	38	18	12	198	2898	0	98	2	130
19	0	0	0	3	3	2	4	2	1	0	1	0	3	2	3	3	1	5	179	0	0	0
20	33	62	5	24	0	91	25	7	267	26	44	70	75	33	37	15	231	223	0	2406	58	160
21	5	19	7	3	0	40	4	3	28	6	2	24	14	11	18	13	13	6	0	21	300	14
22	70	45	1	118	0	7	6	63	32	54	106	7	49	40	36	16	111	182	0	58	7	1000

The final model has an accuracy of 63.6%. The accuracy is rather low for properly identifying users from accelerometer data. However, it is not so low to be useless. It can be used to gather a list of potential users for further investigation should that be warranted. The model can prove useful for helping legal authorities in locating individuals provided they have a warrant.

Ethical Use Policy

It is crucial that ethical procedures be followed at all times in data and machine learning applications. These tasks involve large amounts of data obtained from third parties which should not be taken lightly. The storage of the data should be as secure as possible and in line with legal regulations, cultural expectations, and the rights of privacy afforded to each individual. There are broadly four stages where we can examine the ethical use of this data; the collection stage, the integration and analysis stage, the decision making stage, and the review and revision stage.

The first stage is on data collection. Data should be collected in an ethical manner where all participants are aware of the data that is being collected and have opted into the process. The data was collected by outside organization but was done so in an experimental method where each android user was walking a predefined path, so it is safe to conclude this data has been ethically collected (Casale 2014). Furthermore, data should only be collected as it fits the purpose of the task. This data was collected for the reason of predicting specific users based on accelerometer data and should not include any irrelevant data. This is certainly the case for this data set as it only has four fields, those being: time, x acceleration, y acceleration, and z acceleration (Casale 2014).

Moving onto the second stage integration and analysis is where the data is analyzed. As machine learning models can derive a multitude of information from its training, steps should be taken to ensure user's privacy is safeguarded. When possible data should be anonymized as it is in the data we have sourced. If data turns out to be not fit for purpose it should be discarded and not stored within the company. This limits the ability for misuse of the data and potential for data leaks in the event of breaches (Schwartz 2010).

The third stage is decision making and encompasses how the company will use the developed model, account for potential infringement on users, and assess whether it is in accordance with the law and cultural attitudes (Schwartz 2010). In the case of our analysis and model this would mean providing users methods to reach out in events they believe we have infringed upon their privacy. This would occur in the event that the model is used for tracking user's accelerometer information without just cause. As it stands, the primary use would likely be for assisting police in locating individuals with a warrant, but if it was exercised outside of that capacity there would be grounds for infringement. To further specify that means this model is developed as a form of security and should not be accessible under normal means. Should someone within the company access it to track or identify a user the employee data should be logged and reviewed for each instance. This process would create accountability and ensure that the system is not being abused.

Finally, the fourth stage is to review and revise. Once the model is complete it should be reviewed periodically to ensure its efficacy. Models that are no longer accurate should be discarded or updated to increase their use. On top of maintaining the accuracy of the models, it is important to determine if the model has any unanticipated results. If the model turns out to be accurate in predicting something other than what it was designed for, it should be discarded as the data was not collected from users for that purpose. Likewise, any data which is superficial and does not add to the accuracy of the model in a significant way should be discarded (Schwartz 2010).

References

Casale,Pierluigi. (2014). *User Identification From Walking Activity*. UCI Machine Learning Repository. https://doi.org/10.24432/C5WC8S.

Lantz, Brett. (2019). *Machine Learning with R - Third Edition*. Packt Publishing.

Rodríguez-Martín, D., Samà, A., Pérez-López, C., Català, A., Cabestany, J., & Rodríguez-Molinero, A. (2013). *SVM-based posture identification with a single waist-located triaxial accelerometer*. Expert Systems With Applications, 40(18), 7203–7211. https://doi.org/10.1016/j.eswa.2013.07.028

Schwarts, Paul M. (2010). *Data Protection Law and The Ethical Use of Analytics*. The Centre for Information Policy Leadership.