Asteroid_K-means_and_Decision_Trees

Annika Lin

2023-04-13

```
df <- read.csv("~/Documents/Georgetown/Spring23/Statistical Learning & Data Science/Proj
ect/NASA-asteroid-Classification-master/nasa_4_4_23.csv")
df <- df[ , !(names(df) %in% c("X"))]
```

KNN

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

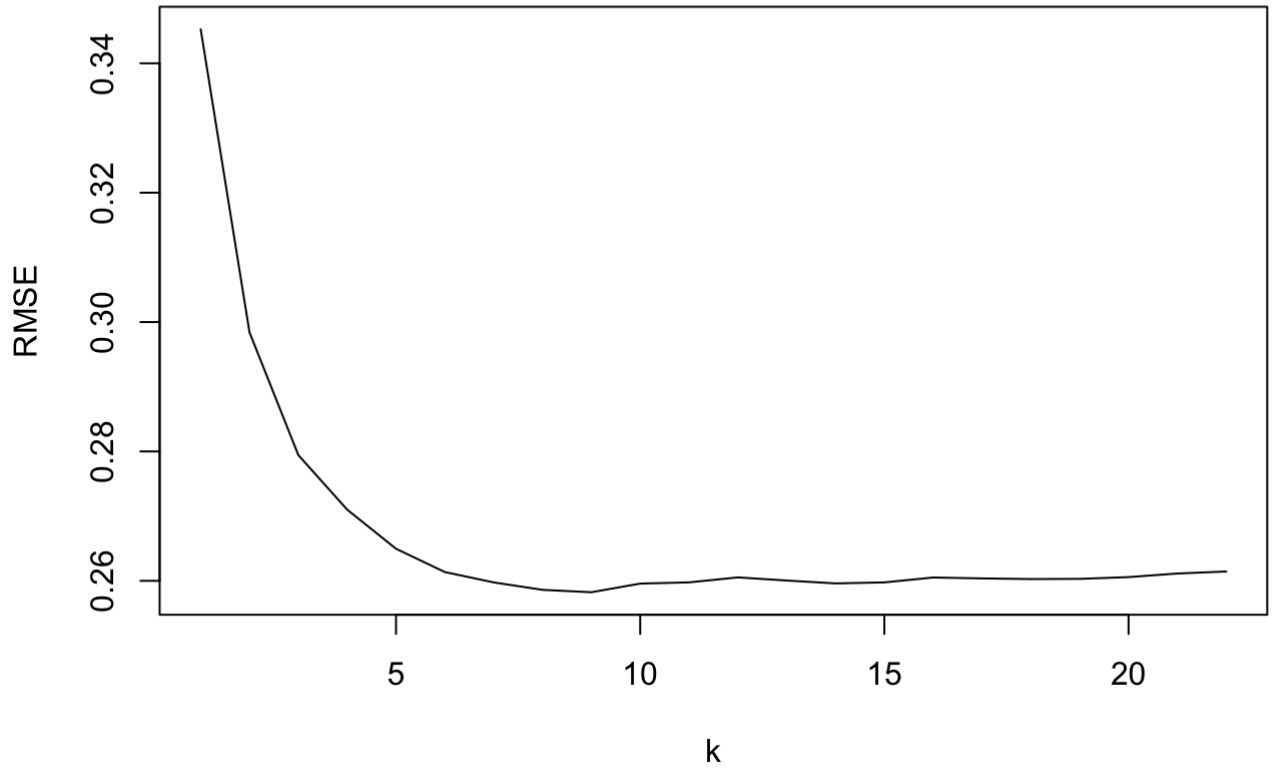
```
df1 <- df
df1$Hazardous <- as.numeric(df1$Hazardous=="True")

set.seed(1)
nasa.knn <- train(Hazardous ~ .,
method = "knn",
tuneGrid = expand.grid(k = 1:22),
trControl = trainControl(method="cv", number=10,
savePredictions = TRUE),
preProcess = c("center", "scale"),
metric = "RMSE",
data = df1)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
nasa.knn
```

```
## k-Nearest Neighbors
##
## 3079 samples
##   20 predictor
##
## Pre-processing: centered (20), scaled (20)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2772, 2771, 2771, 2771, 2771, 2771, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    1  0.3452597  0.2299425  0.1195133
##    2  0.2984023  0.3142152  0.1240099
##    3  0.2794311  0.3667528  0.1254720
##    4  0.2710007  0.3966877  0.1278470
##    5  0.2649554  0.4245425  0.1277755
##    6  0.2613555  0.4425494  0.1282852
##    7  0.2597581  0.4513895  0.1294344
##    8  0.2586058  0.4592961  0.1309074
##    9  0.2582363  0.4632208  0.1323350
##   10  0.2595728  0.4595391  0.1344121
##   11  0.2597619  0.4603941  0.1356804
##   12  0.2605296  0.4586126  0.1371685
##   13  0.2600484  0.4629537  0.1378296
##   14  0.2596005  0.4664108  0.1386083
##   15  0.2597583  0.4690749  0.1394879
##   16  0.2605140  0.4669937  0.1408594
##   17  0.2603744  0.4699129  0.1412371
##   18  0.2602734  0.4733221  0.1416620
##   19  0.2603030  0.4752344  0.1421008
##   20  0.2605757  0.4757612  0.1428261
##   21  0.2611299  0.4741765  0.1436628
##   22  0.2614439  0.4725970  0.1442686
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```

```
plot(nasa.knn$results[,1:2], type="l")
```



Let us compare the results with fitting a logistic regression model using 10-fold CV

```
#UPDATE : variables selected previously with LASSO

set.seed(1)
  fit.nasa <- train(as.factor(Hazardous) ~ Absolute.Magnitude+Est.Dia.in.KM.min.+Orbit.U
ncertainity+Minimum.Orbit.Intersection+Inclination+Mean.Motion+Range.Dia.in.KM,
  method="glm", family="binomial",
  trControl = trainControl(method="cv", number=10,
  savePredictions = TRUE,
  classProbs = TRUE),
  metric = "Accuracy",
  data=df)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
fit.nasa
```

```
## Generalized Linear Model
##
## 3079 samples
##    7 predictor
##    2 classes: 'False', 'True'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2771, 2771, 2772, 2770, 2771, 2772, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9600511  0.8346053
```

Comparing KNN and Logistic model

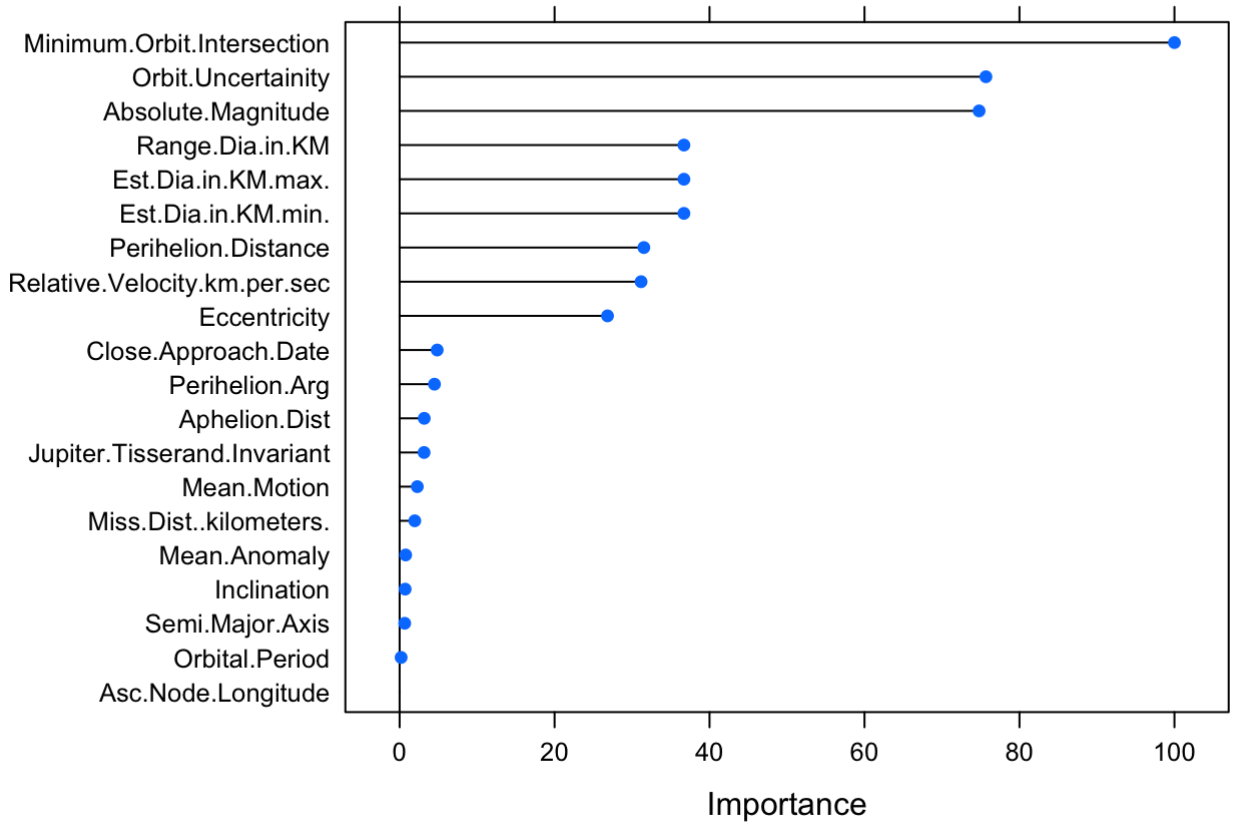
```
varImp(nasa.knn)
```

```
## loess r-squared variable importance
##
##                              Overall
## Minimum.Orbit.Intersection   100.0000
## Orbit.Uncertainity            75.6704
## Absolute.Magnitude            74.7904
## Range.Dia.in.KM               36.6967
## Est.Dia.in.KM.max.            36.6967
## Est.Dia.in.KM.min.            36.6967
## Perihelion.Distance           31.5253
## Relative.Velocity.km.per.sec  31.1675
## Eccentricity                  26.8346
## Close.Approach.Date            4.8503
## Perihelion.Arg                 4.4967
## Aphelion.Dist                  3.1796
## Jupiter.Tisserand.Invariant    3.1586
## Mean.Motion                    2.2835
## Miss.Dist..kilometers.         1.9569
## Mean.Anomaly                   0.7811
## Inclination                    0.7153
## Semi.Major.Axis                0.6577
## Orbital.Period                 0.1959
## Asc.Node.Longitude             0.0000
```

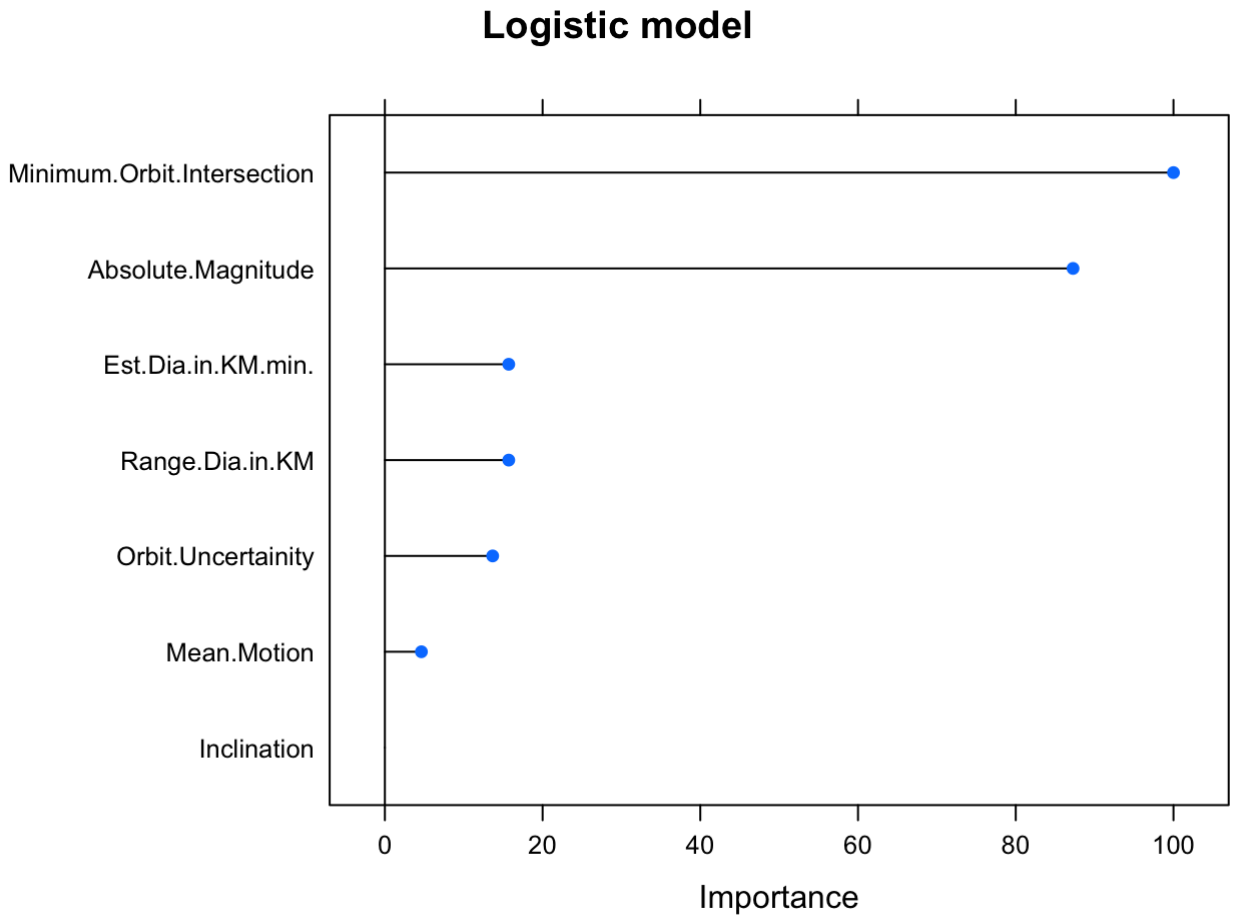
```
varImp(fit.nasa)
```

```
## glm variable importance
##
##                            Overall
## Minimum.Orbit.Intersection 100.000
## Absolute.Magnitude          87.263
## Est.Dia.in.KM.min.          15.708
## Range.Dia.in.KM             15.708
## Orbit.Uncertainity          13.666
## Mean.Motion                  4.632
## Inclination                  0.000
```

```
plot(varImp(nasa.knn), main="kNN variable importance")
```




```
plot(varImp(fit.nasa), main="Logistic model")
```



Decision Tree

```
df <- read.csv("~/Documents/Georgetown/Spring23/Statistical Learning & Data Science/Proj
ect/NASA-asteroid-Classification-master/nasa_4_4_23.csv")
df <- df[ , !(names(df) %in% c("X"))]
```

```
library(rpart)
```

```
set.seed(1)
df.CVrpart <- train(Hazardous ~ ., data=df,
method="rpart",
tuneGrid = expand.grid(cp = seq(0.005, 0.05, length=10)),
trControl = trainControl(method = "cv", number=10,
savePredictions = TRUE,
selectionFunction = "oneSE") )
df.CVrpart
```

```
## CART
##
## 3079 samples
##   20 predictor
##    2 classes: 'False', 'True'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2771, 2771, 2772, 2770, 2771, 2772, ...
## Resampling results across tuning parameters:
##
##   cp     Accuracy  Kappa
##   0.005  0.993829  0.9743262
##   0.010  0.993829  0.9743262
##   0.015  0.993829  0.9743262
##   0.020  0.993829  0.9743262
##   0.025  0.993829  0.9743262
##   0.030  0.993829  0.9743262
##   0.035  0.993829  0.9743262
##   0.040  0.993829  0.9743262
##   0.045  0.993829  0.9743262
##   0.050  0.993829  0.9743262
##
## Accuracy was used to select the optimal model using  the one SE rule.
## The final value used for the model was cp = 0.05.
```

```
print(df.CVrpart$finalModel)
```

```
## n= 3079
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 3079 435 False (0.858720364 0.141279636)
##   2) Absolute.Magnitude>=22.05 1614   7 False (0.995662949 0.004337051) *
##   3) Absolute.Magnitude< 22.05 1465 428 False (0.707849829 0.292150171)
##     6) Minimum.Orbit.Intersection>=0.05013445 1029   0 False (1.000000000 0.00000000
0) *
##     7) Minimum.Orbit.Intersection< 0.05013445 436   8 True (0.018348624 0.981651376)
*
```

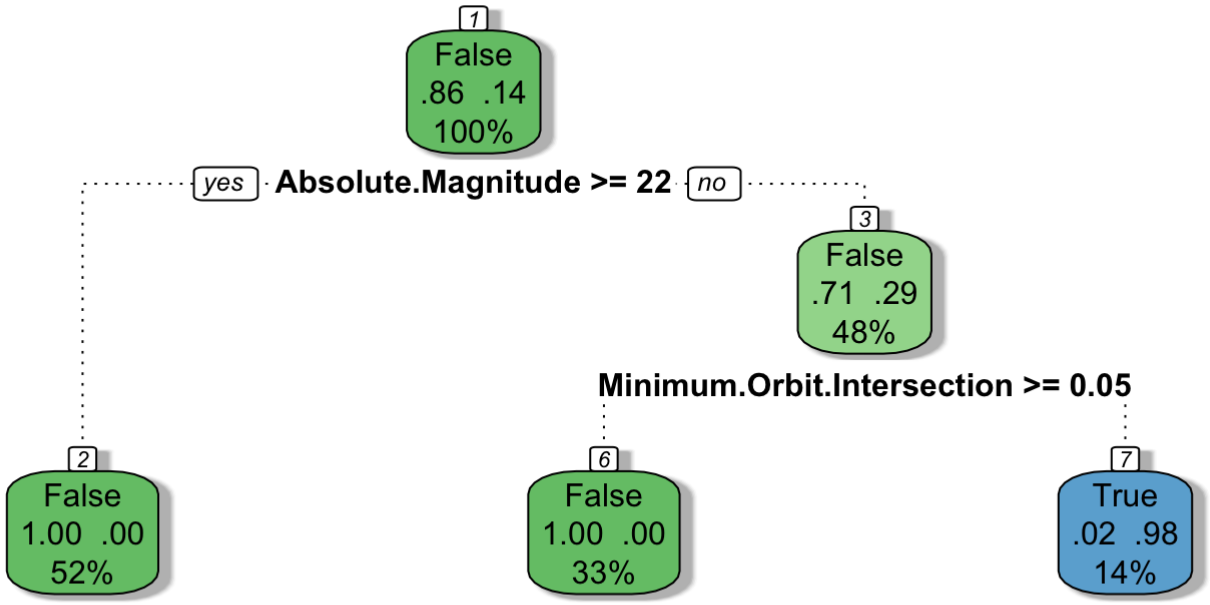
```
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
fancyRpartPlot(df.CVrpart$finalModel)
```




```
varImp(df.CVrpart)
```

```
## rpart variable importance
##
##                              Overall
## Minimum.Orbit.Intersection   100.000
## Absolute.Magnitude            18.004
## Range.Dia.in.KM               18.004
## Est.Dia.in.KM.max.            18.004
## Est.Dia.in.KM.min.            18.004
## Perihelion.Distance           12.614
## Miss.Dist..kilometers.         8.894
## Inclination                    6.351
## Relative.Velocity.km.per.sec   2.235
## Mean.Anomaly                   0.000
## Aphelion.Dist                  0.000
## Close.Approach.Date            0.000
## Eccentricity                   0.000
## Jupiter.Tisserand.Invariant    0.000
## Orbit.Uncertainity             0.000
## Mean.Motion                    0.000
## Asc.Node.Longitude             0.000
## Orbital.Period                 0.000
## Perihelion.Arg                 0.000
## Semi.Major.Axis                0.000
```

Note Absolute Magnitude and Est.Dia and two other variables have the variable importance.

```
head(df.CVrpart$pred)
```

```
##    pred   obs rowIndex    cp Resample
## 1 False False       11 0.005   Fold01
## 2 False False       13 0.005   Fold01
## 3 False False       34 0.005   Fold01
## 4 False False       38 0.005   Fold01
## 5  True  True       66 0.005   Fold01
## 6  True  True       68 0.005   Fold01
```