Module 3 Application of First Method

KNN Predictor

# Project Task Status

**Project proposal and development of the first prediction model KNN is completed.**

# Executive Summary

The objective of this project is to predict the amount of gold spent per player in a DOTA2 game by applying a K-Nearest Neighbors classifier algorithm on a selection of in-game data about the match and the player. The model was successfully developed and has an accuracy of over 90% in predicting the expenditure category of the player (low, medium, high).

# KNN Prediction of Gold Spent: Introduction

In this project, a KNN model will be developed to predict the amount of gold spent per player in a DOTA2 game using various other variables about the player or the specific match. Players will be classified into one of three spending categories - low, medium, and high - based on the distribution of the gold spent variable.

## Implementation Approach: Data Import and Preparation

**library**(readr)  
match <- **read\_csv**('match.csv')

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## match\_id = col\_double(),  
## start\_time = col\_double(),  
## duration = col\_double(),  
## tower\_status\_radiant = col\_double(),  
## tower\_status\_dire = col\_double(),  
## barracks\_status\_dire = col\_double(),  
## barracks\_status\_radiant = col\_double(),  
## first\_blood\_time = col\_double(),  
## game\_mode = col\_double(),  
## radiant\_win = col\_logical(),  
## negative\_votes = col\_double(),  
## positive\_votes = col\_double(),  
## cluster = col\_double()  
## )

ratings <- **read\_csv**('player\_ratings.csv')

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## account\_id = col\_double(),  
## total\_wins = col\_double(),  
## total\_matches = col\_double(),  
## trueskill\_mu = col\_double(),  
## trueskill\_sigma = col\_double()  
## )

players <- **read\_csv**('players.csv')

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_double(),  
## stuns = col\_character(),  
## unit\_order\_none = col\_logical(),  
## unit\_order\_taunt = col\_logical(),  
## unit\_order\_cast\_rune = col\_logical(),  
## unit\_order\_patrol = col\_logical(),  
## unit\_order\_vector\_target\_position = col\_logical(),  
## unit\_order\_radar = col\_logical(),  
## unit\_order\_set\_item\_combine\_lock = col\_logical(),  
## unit\_order\_continue = col\_logical()  
## )  
## ℹ Use `spec()` for the full column specifications.

## Warning: 15 parsing failures.  
## row col expected actual file  
## 2288 unit\_order\_cast\_rune 1/0/T/F/TRUE/FALSE 1.0 'players.csv'  
## 14401 unit\_order\_none 1/0/T/F/TRUE/FALSE 1.0 'players.csv'  
## 94568 unit\_order\_cast\_rune 1/0/T/F/TRUE/FALSE 1.0 'players.csv'  
## 118205 unit\_order\_cast\_rune 1/0/T/F/TRUE/FALSE 1.0 'players.csv'  
## 130076 unit\_order\_none 1/0/T/F/TRUE/FALSE 2.0 'players.csv'  
## ...... .................... .................. ...... .............  
## See problems(...) for more details.

The datasets need to be merged into one master dataset with the following columns from each: - Match: Duration, first blood time - Ratings: total matches - Match: kills, deaths, gold per minute (the target variable)

Ratings and players will be merged on account\_id while match and players will be merged on match\_id.

match\_keep <- match[,**c**('match\_id', 'duration', 'first\_blood\_time')]  
ratings\_keep <- ratings[,**c**('account\_id', 'total\_matches')]  
players\_keep <- players[,**c**('match\_id', 'account\_id', 'gold\_per\_min', 'kills', 'deaths', 'gold\_spent')]  
  
m1 <- **merge**(match\_keep, players\_keep, by = 'match\_id', all = FALSE)  
*# At the second merge, R runs out of memory so only the first 1,000 entries will be used*  
m1\_short <- **head**(m1, 1000)  
players\_keep\_short <- **head**(players\_keep, 300000)  
df <- **merge**(m1\_short, players\_keep\_short, by = 'account\_id', all = FALSE)  
dota <- df[**sample**(**nrow**(df), 1000), ]  
drops <- **c**('account\_id', 'match\_id.x', 'match\_id.y', 'gold\_per\_min.y', 'kills.y', 'deaths.y', 'gold\_spent.y')  
dota <- dota[, **!**(**names**(dota) **%in%** drops)]

At this point, we have a dataset with 1,000 randomly selected matches that will be used in the development of a KNN model. The final step of data preparation is creating a categorical variable for classification based on the gold spent variable. This will break spending into a low, medium, and high group.

**hist**(dota**$**gold\_spent.x)



*# Categories: 0-10000, 10001-20000, 20001+*  
  
**library**(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

dota <- **mutate**(dota, spend\_cat = **case\_when**(gold\_spent.x **<=** 10000 **~** 0,gold\_spent.x **>** 10000 **&** gold\_spent.x **<=** 20000 **~** 1, gold\_spent.x **>** 20000 **~** 2))

## Preparing for the KNN Model

In order to run a KNN classifier, the data must first be normalized. This ensures that differences in range or scale between variables does not have an undue impact on the distances calculated between them.

*# Normalize*  
nor <- **function**(x) {(x**-min**(x)) **/** (**max**(x)**-** **min**(x))}  
dota\_norm <- **as.data.frame**(**lapply**(dota[,**c**(1,2,3,4,5,6)], nor))  
**summary**(dota\_norm)

## duration first\_blood\_time gold\_per\_min.x kills.x   
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2295 1st Qu.:0.06569 1st Qu.:0.2656 1st Qu.:0.1154   
## Median :0.3428 Median :0.29015 Median :0.3804 Median :0.2308   
## Mean :0.3726 Mean :0.32553 Mean :0.4075 Mean :0.2864   
## 3rd Qu.:0.5288 3rd Qu.:0.49270 3rd Qu.:0.5141 3rd Qu.:0.3846   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## deaths.x gold\_spent.x   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2500 1st Qu.:0.1853   
## Median :0.4000 Median :0.2923   
## Mean :0.4013 Mean :0.3196   
## 3rd Qu.:0.5500 3rd Qu.:0.4233   
## Max. :1.0000 Max. :1.0000

Next, the data will be broken into a 75/25 training and test split. This will allow for eventual calculation of an accuracy of the KNN model.

*# 25/75 Test/Train Split*  
ran <- **sample**(1**:nrow**(dota\_norm), .75 **\*** **nrow**(dota\_norm))  
dota\_train <- dota\_norm[ran,]  
dota\_test <- dota\_norm[**-**ran,]  
  
*# Extract 14th column as the response*  
dota\_target\_cat <- dota[ran,7]  
dota\_test\_cat <- dota[**-**ran, 7]

## KNN Model Development

Finally, the KNN model can be built. A K of 10 will be used. The accuracy of the classifier will also be determined using a confusion matrix.

**library**(class)  
pr <- **knn**(dota\_train, dota\_test, cl = dota\_target\_cat, k = 10)  
tab <- **table**(pr, dota\_test\_cat)  
  
accuracy <- **function**(x) {**sum**(**diag**(x) **/** (**sum**(**rowSums**(x)))) **\*** 100}  
**accuracy**(tab)

## [1] 90.8

## Data Analysis and Results

The model has an accuracy of 90.4% in predicting the amount of gold a player will spend using match duration, first blood time, gold per minute, player kills, and player deaths.

## Discussion

The best model in this analysis was the KNN model which resulted in a 90.4% accuracy in predicting whether a player would be in a low, medium, or high expenditure category in a specific match. Therefore, this model is recommended. One limitation, however, is that it is not a continuous predictor and can only classify players into spending groups without predicting the actual amounts that will be spent.