Kobe Bryant Shot Selection

Kurt Doyle, Shiva Duddupudi,
Ruxin Li, Wade Strain

Kaggle Competition - Which Shots did Kobe Make?

Overview: The data contains the location and circumstances of every field goal attempted by Kobe Bryant during his 20-year career.

Objective: Binary Classification ; Given a held out test set, predict whether the basket went in.

Data: 25 variables

Type of shot, location information, time information, season information,

game_event_id	game_id	lat	loc_x	loc_y	lon	minutes_remaining	period	playoffs	season	seconds_remaining	shot_distance	shot_made_flag	shot_type	shot_zone_area	shot_zone_basic	shot_zone_range	team_id	team_name	game_date	matchup	opponent	shot_id
10	20000012	33.9723	167	72	-118.103	10	1	0	2000-01	27	18		2PT Field Go	Right Side(R)	Mid-Range	16-24 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	1
12	20000012	34.0443	-157	0	-118.427	10	1	0	2000-01	22	15	0	2PT Field Go	Left Side(L)	Mid-Range	8-16 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	2
35	20000012	33.9093	-101	135	-118.371	7	1	0	2000-01	45	16	1	2PT Field Go	Left Side Center(LC	Mid-Range	16-24 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	3
43	20000012	33.8693	138	175	-118.132	6	1	0	2000-01	52	22	0	2PT Field Go	Right Side Center(I	Mid-Range	16-24 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	4
155	20000012	34.0443	0	0	-118.27	6	2	0	2000-01	19	0	1	2PT Field Go	Center(C)	Restricted Area	Less Than 8 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	5
244	20000012	34.0553	-145	-11	-118.415	9	3	0	2000-01	32	14	0	2PT Field Go	Left Side(L)	Mid-Range	8-16 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	6
251	20000012	34.0443	0	0	-118.27	8	3	0	2000-01	52	0	1	2PT Field Go	Center(C)	Restricted Area	Less Than 8 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	7
254	20000012	34.0163	1	28	-118.269	8	3	0	2000-01	5	2		2PT Field Go	Center(C)	Restricted Area	Less Than 8 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	8
265	20000012	33.9363	-65	108	-118.335	6	3	0	2000-01	12	12	1	2PT Field Go	Left Side(L)	In The Paint (Non-F	8-16 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	9
294	20000012	33.9193	-33	125	-118.303	3	3	0	2000-01	36	12	0	2PT Field Go	Center(C)	In The Paint (Non-F	8-16 ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	10
309	20000012	33.8063	-94	238	-118.364	1	3	0	2000-01	56	25	0	3PT Field Go	Left Side Center(LC	Above the Break 3	24+ ft.	1.61E+09	Los Angeles La	10/31/2000	LAL @ POR	POR	11
4	20000019	33.9173	121	127	-118.149	11	1	0	2000-01	0	17	1	2PT Field Go	Right Side Center(I	Mid-Range	16-24 ft.	1.61E+09	Los Angeles La	11/1/2000	LAL vs. UTA	UTA	12
27	20000019	33.9343	-67	110	-118.337	7	1	0	2000-01	9	12	1	2PT Field Go	Left Side(L)	In The Paint (Non-F	8-16 ft.	1.61E+09	Los Angeles La	11/1/2000	LAL vs. UTA	UTA	13

Notebook Critique: SVM with Radial Kernel

Positive: Good use of augmenting the data. Created a time remaining column which combined two other columns.

Negative: Limited use of features (three of them), predicted actual class

Result: Kaggle Score 13.46328

```
#handle with the train features
train$shot_distance[train$shot_distance>40] <- 40
train$time_remaining <- train$minutes_remaining*60+train$seconds_remaining;
```

```
#build svm model by train data
wts=c(1,1)
names(wts)=c(1,0)
model <- svm(shot_made_flag~., data=dat, kernel="radial",  gamma=1, cost=1, class.weights=wts)
```

https://www.kaggle.com/shuicaorj1036081/predict/code

Notebook Critique: XGBoost

Positive: Using approximate greedy algorithm when data is large, fits regression tree to residuals; Parallelization of tree construction using all of your CPU cores during training.

Negative: Easy overfitting and it is not good at extrapolating unseen values

Result: Kaggle Score 0.62363

```
train <- as.matrix(train)
test <- as.matrix(test)
```

```
myxgb <- xgboost(data=train, label=labels, objective="binary:logistic",
    nround=149, eta=0.35, maxdepth=10, gamma=50, subsample=0.6, colsample_bytree=0.5)
```

https://www.kaggle.com/amineteffal/script-aminos-kb

Team Model Exploration

Model	Log Loss
XGBoost	0.64997
SVM - Polynomial	0.66780
SVC - Linear	0.66797
Random Forest	0.67752
Ensemble	0.67970
50 / 50 Benchmark	0.69314

Team Model: Stacked Ensemble

Utilized 7 features

Used cross validation with five folds

4 different models: glm, knn, rpart, svm-radial

Measured model correlation to make sure the model were not too similar

Kaggle Score: 0.67970

```
# Stacking Algorithm
control <- trainControl(method='cv', number=5, savePredictions='final')
algList <- c('glm','knn','rpart','svmRadial')
set.seed(2)
models <- caretList(shot_made_flag~., data=Train, trControl=control, methodList=algList)

results <- resamples(models)
summary(results)
modelCor(results)

stack.rf <- caretStack(models, method='rf', metric='LogLoss', trControl=control)
print(stack.rf)

test.x = test[, -which(names(test) %in% 'shot_made_flag')]
pred.y <- predict(stack.rf, newdata=test.x)
```

Team Model: Classification with Random Forest

Utilized 9 features

Varied *mtry* and *ntrees* values in a loop to calculate the overall error rate using random forest.

Best Results: mtry = 3 ; ntrees = 200 ; Kaggle Score: 0.67752

```
for (i in 1:length(mtry.num)){
  for (j in 1:length(trees.num)){
    rf.shots <- randomForest(shot_made_flag ~., data = x_train, mtry = i, ntree = j, importance = FALSE)
    yhat.rf <- predict(rf.shots, newdata = x_test)
    rf.table <- table(y_test, yhat.rf)
    rf.errorRate <- (rf.table[1,2] + rf.table[2,1]) / sum(rf.table)
    error.rate[k] <- rf.errorRate
    mt[k] <- mtry.num[i]
    tr[k] <- trees.num[j]
    print(k)
    k <- k + 1
  }
}
```

Team Model: XGBoost

Utilized 7 features

Used a gbtree booster while evaluating log loss, across several parameters

Kaggle Score: 0.64997

```
params <- list(booster = 'gbtree',
               objective = 'binary:logistic',
               eval_metric = 'logloss',
               eta = 0.3,
               gamma = 0,
               max_depth = 6,
               min_child_weight = 1,
               subsample = 1,
               colsample_bytree = 1)
```