Kobe Bryant Shot Selection

統研碩一 郭又嘉

數據名稱-預測Kobe是否進球

- action_type
  combined_shot_type
  game_event_id
  game_id
  lat
  loc_x
  loc_y
  lon
  minutes_remaining
  period
  playoffs
  season
  seconds_remaining

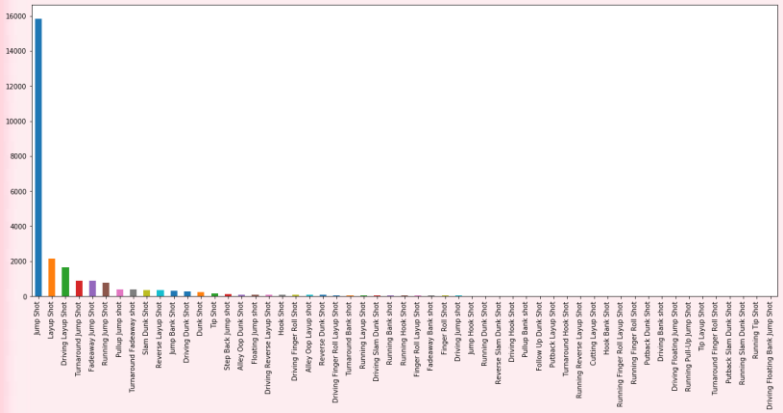
- shot_distance
  shot_made_flag
  shot_type
  shot_zone_area
  shot_zone_basic
  shot_zone_range
  team_id
  team_name
  game_date
  matchup
  opponent
  shot_id

資料整理

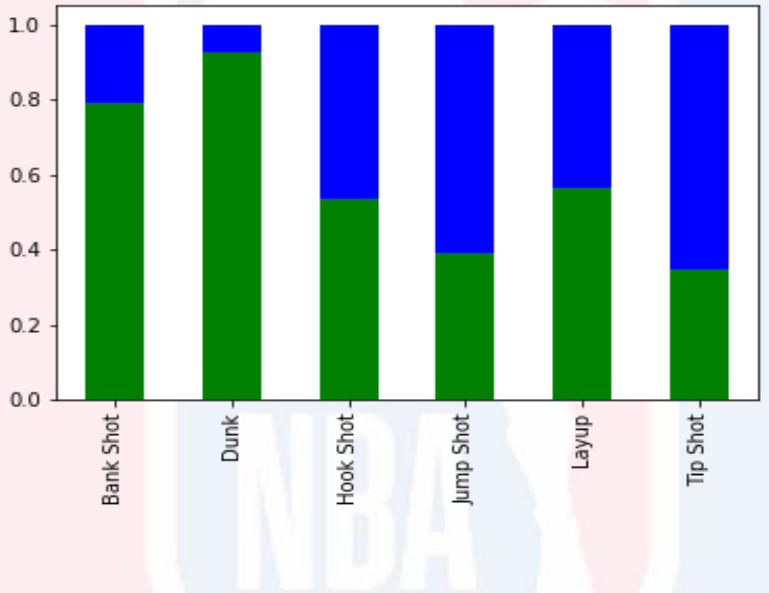
挑選對"進球"有用的資料作使用

挑選資料

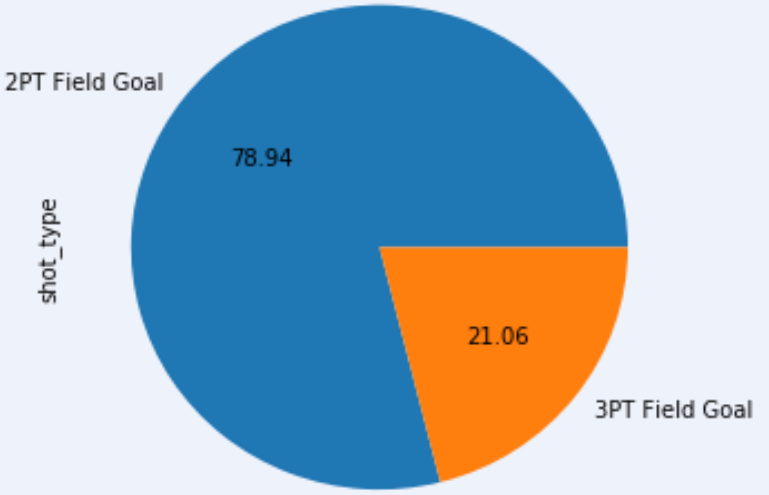
- 查看資料 (action_type、combined_shot_type、shot_type)



action_type



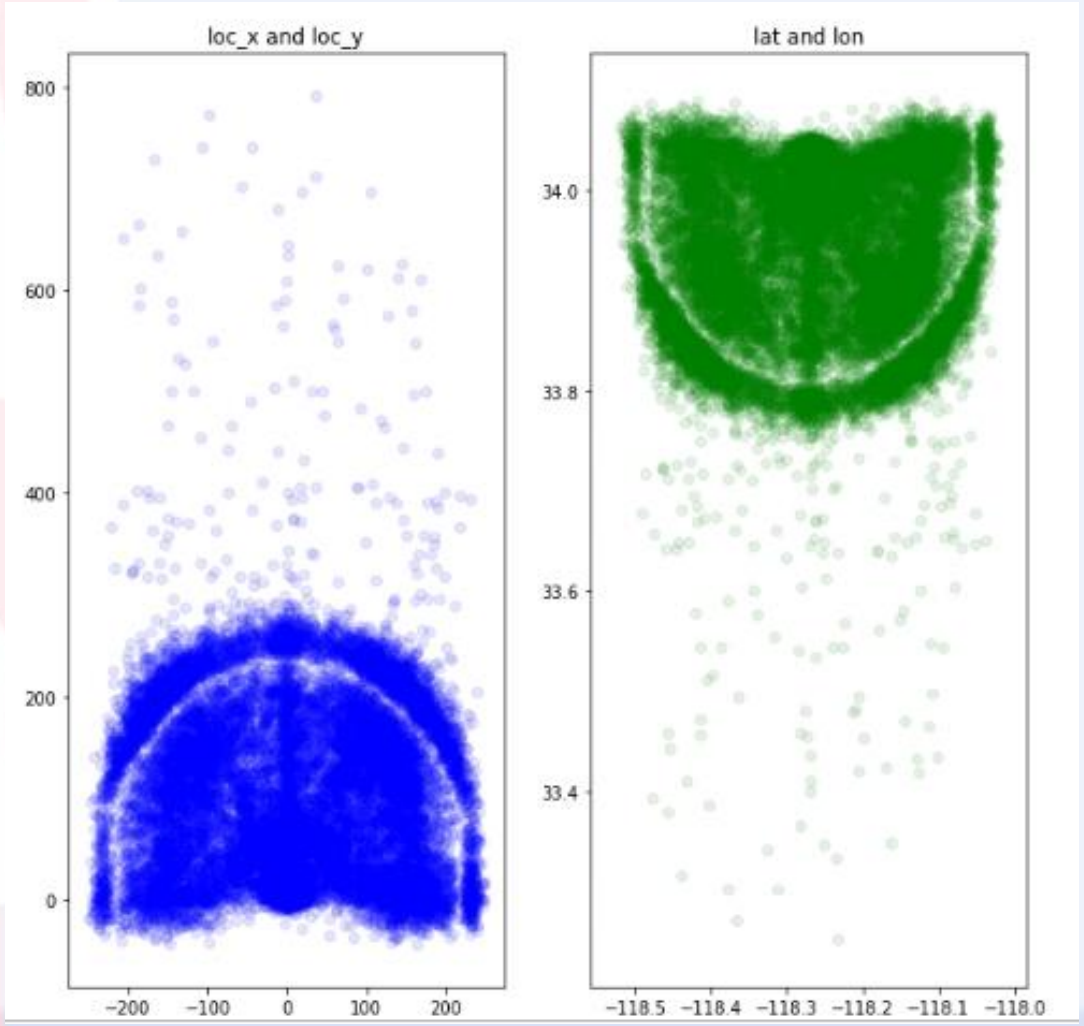
combined_shot_type



shot_type

挑選資料

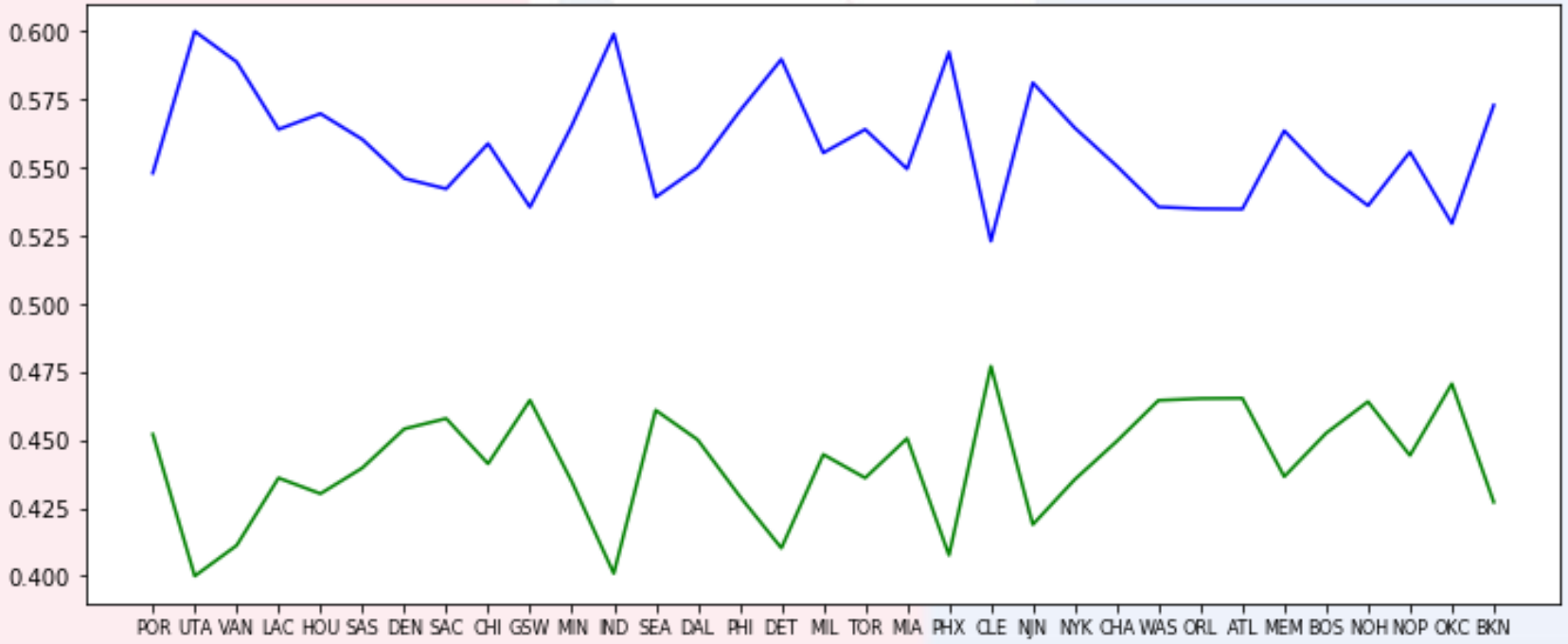
- 查看資料 (loc_x 、loc_y、lat、lon)
- 球場是半圓形的，把座標位置轉換為極坐標來表示
- 新增dist、angle變數
- 有加角度的 dist 和本來資料裡的 shot_distance 為正相關，選擇使用 dist



挑選資料

	matchup	opponent
1	LAL @ POR	POR
2	LAL @ POR	POR
3	LAL @ POR	POR
4	LAL @ POR	POR
5	LAL @ POR	POR
6	LAL @ POR	POR
8	LAL @ POR	POR
9	LAL @ POR	POR
10	LAL @ POR	POR

- 查看資料 (matchup、opponent)
- 兩者資料類型相似，選擇使用opponent，碰到不一樣的對手也會影響進球的狀態，所以要考慮進去



挑選資料

- 查看資料 (minutes_remaining、seconds_remaining)
- 把剩餘時間做合併 (單位：秒)

- 查看資料 (season)
- 將賽季轉為用賽季的年份來代表，而每個時期的狀態不一樣，所以要考慮進去

```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
       2011, 2012, 2013, 2014, 2015, 1996, 1997, 1998, 1999], dtype=int64)
```

挑選資料

(1) 1996-1999 新人期，從高中球隊到進入NBA，成為史上最年輕的先發球員

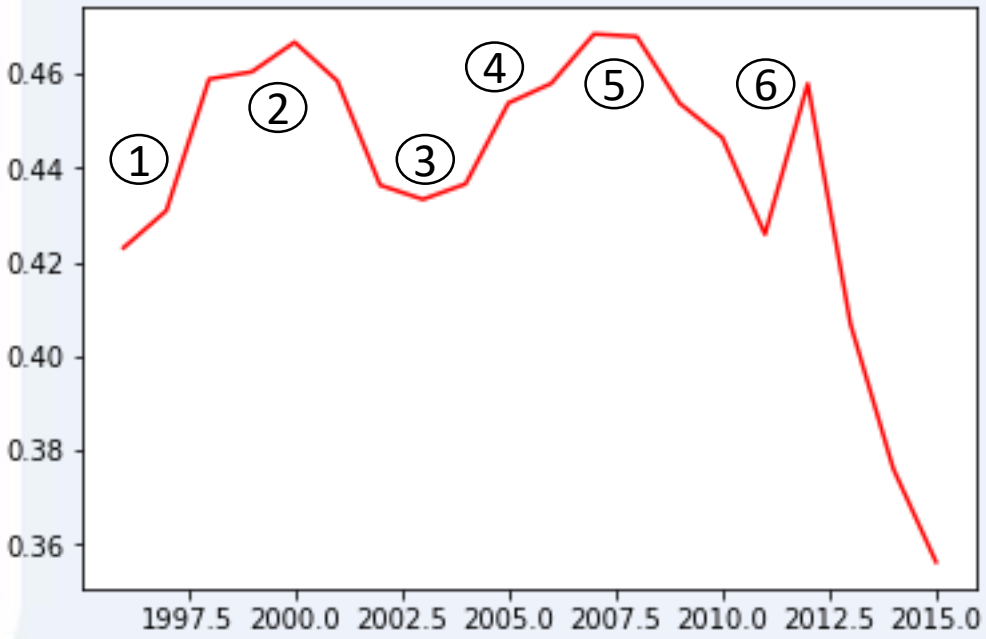
(2) 1999-2003 高峰期，湖人隊連奪三次NBA總冠軍，與其他明星球員被譽為「四大天王」

(3) 2003-2004 低潮期，遇上性侵事件和球員內鬥，讓此球季為柯比生涯的最低點

(4) 2004-2007 轉折期，球隊好轉，柯比在此季創下湖人隊的單季最高個人得分的球隊紀錄

(5) 2007-2013 巔峰期，獲得多項殊榮：NBA最有價值球員獎項、多次總決賽MVP.....等等

(6) 2013-2016 大傷至退休時期



彙整資料

1. 先刪除shot_made_flag中有nan的值

2. 刪除名稱與其他無用數據

```
drops = ['shot_id', 'team_id', 'team_name', 'shot_zone_area', 'shot_zone_range', 'shot_zone_basic', \
         'matchup', 'lon', 'lat', 'seconds_remaining', 'minutes_remaining', \
         'shot_distance', 'loc_x', 'loc_y', 'game_event_id', 'game_id', 'game_date','playoffs']
for drop in drops:
    nona = nona.drop(drop, axis=1)
nona.head()
```

3. 將類別資料轉為數值資料

   action_type、combined_shot_type

   shot_type、opponent、period、season

資料分析

用兩種方法來預測是否進球

分析方法

1. LogisticRegression

2. LinearDiscriminantAnalysis (線性判別分析)

- LDA將數據點投影到維度更低的空間中，使投影後的點會按照類別區分
- LDA目標是讓不同類別之間的距離越遠越好，同一類別的距離越近越好

分析方法結果比較

LogisticRegression

```
from sklearn.linear_model import LogisticRegression
from sklearn.grid_search import GridSearchCV

clf = LogisticRegression(penalty='l2', max_iter=1000)
gs = GridSearchCV(clf,param_grid = grid, scoring='roc_auc', cv=fold)
gs.fit(X_train, y_train)
print("所有C值下之結果：",gs.grid_scores_)
print("最好的C值：",gs.best_params_)
print("LR模型準確度：",gs.best_score_)
```

LinearDiscriminantAnalysis

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

lda = LinearDiscriminantAnalysis()
lda_grid = GridSearchCV(lda,param_grid = {'solver': ['lsqr'],'shrinkage': [0, 0.25, 0.5, 0.75, 1],
                                          'n_components': [None,2,5,10]}, cv = fold)

lda_grid.fit(X_train, y_train)

print("最好的參數：",lda_grid.best_params_)
print("LDA模型準確度：",lda_grid.best_score_)
```

準確度

```
print("LR準確度：",gs.score(X_test,y_test))
print("LDA準確度：",lda_grid.score(X_test,y_test))
```

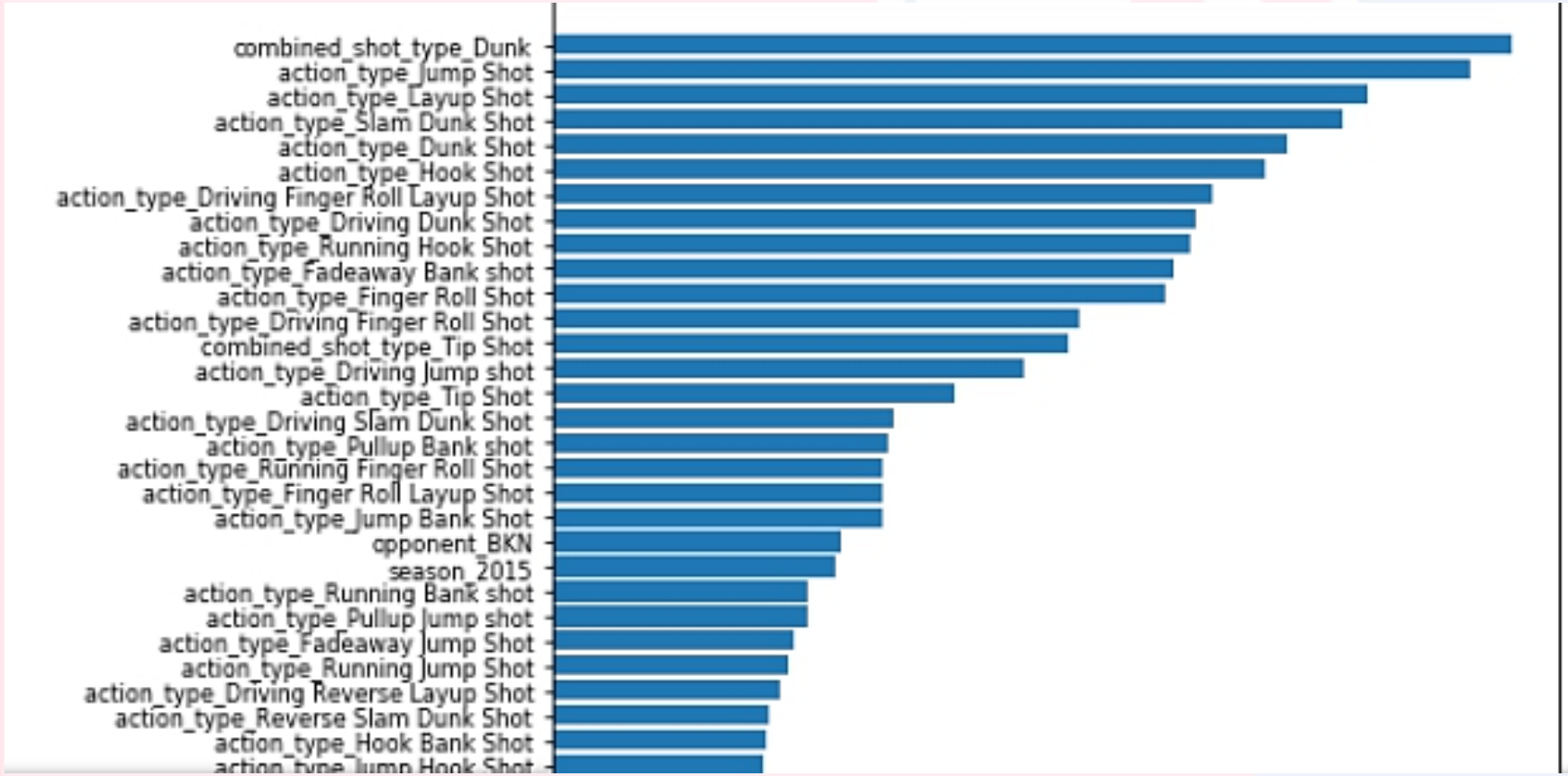
```
LR準確度： 0.684317096528529
LDA準確度： 0.6806980308925834
```

資料分析

找關於Kobe進球的重要特徵

Important features

- 以迴歸係數來做標準，看哪個特徵影響最大



- 最主要影響進球的前三名特徵：
- **combined_shot_type_Dunk**
- **action_type_Jump Shot**
- **action_type_Layup Shot**

- 前二十名都是以"投球的動作"為主要的影響

Important features

```
出手率：
Jump Shot     0.672307
Layup         0.223375
Dunk          0.085478
Bank Shot     0.008286
Hook Shot     0.005931
Tip Shot      0.004623
Name: combined_shot_type, dtype: float64
命中率：
Bank Shot     0.791667
Dunk          0.928030
Hook Shot     0.535433
Jump Shot     0.391071
Layup         0.565093
Tip Shot      0.348684
Name: combined_shot_type, dtype: float64
```

- 出手前三名是Jump Shot(跳投)、Layup(上籃)、Dunk(灌籃)
- 命中率前三名是Dunk、Bank Shot(擦板投籃)、Layup
- 距離跟命中率成反比，投籃命中率隨著籃框的距離減少而增加
- Jump Shot是Kobe出手率高但失誤率也高的 (可以從這防守)

END