Competition1

指導教授：李政德　教授

組員:王子豪 H24031346 周逸平 H24036087

一、模型介紹(Description)

我們分別使用了隨機森林模型（Random Forest Model)、邏輯式迴歸模型（Logistic Regression Model)、深度神經網路模型（Deep Neural Network Model）對 Kobe Bryant 的投籃類型進行預測，之後使用投票的方式，從三個模型預測結果中採用多數決以此決定最終預測結果。若三者意見相左以邏輯式迴歸模型預測結果為基準。

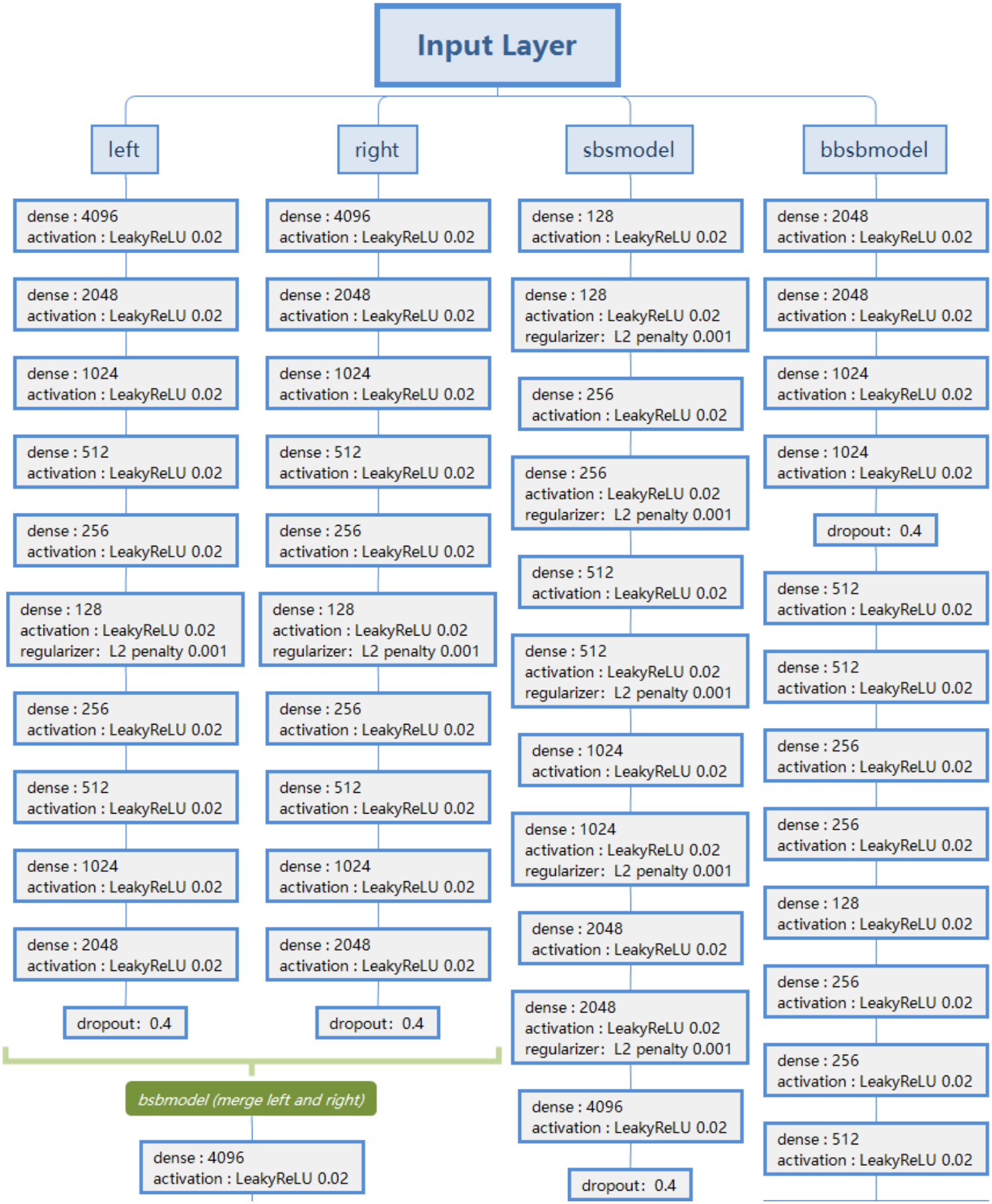
1. 資料前處理：
   a. 將 game_date 變數轉換為浮點數型態，以此將其視作連續型變數。
   b. 將 loc_x，loc_y 此二變數取絕對值，此外將 playoffs，period，season ， shot_made_flag ， shot_zone_area ， opponent ， action_type 等變數視作類別變數並轉換成虛擬變數（Dummy Variable)。

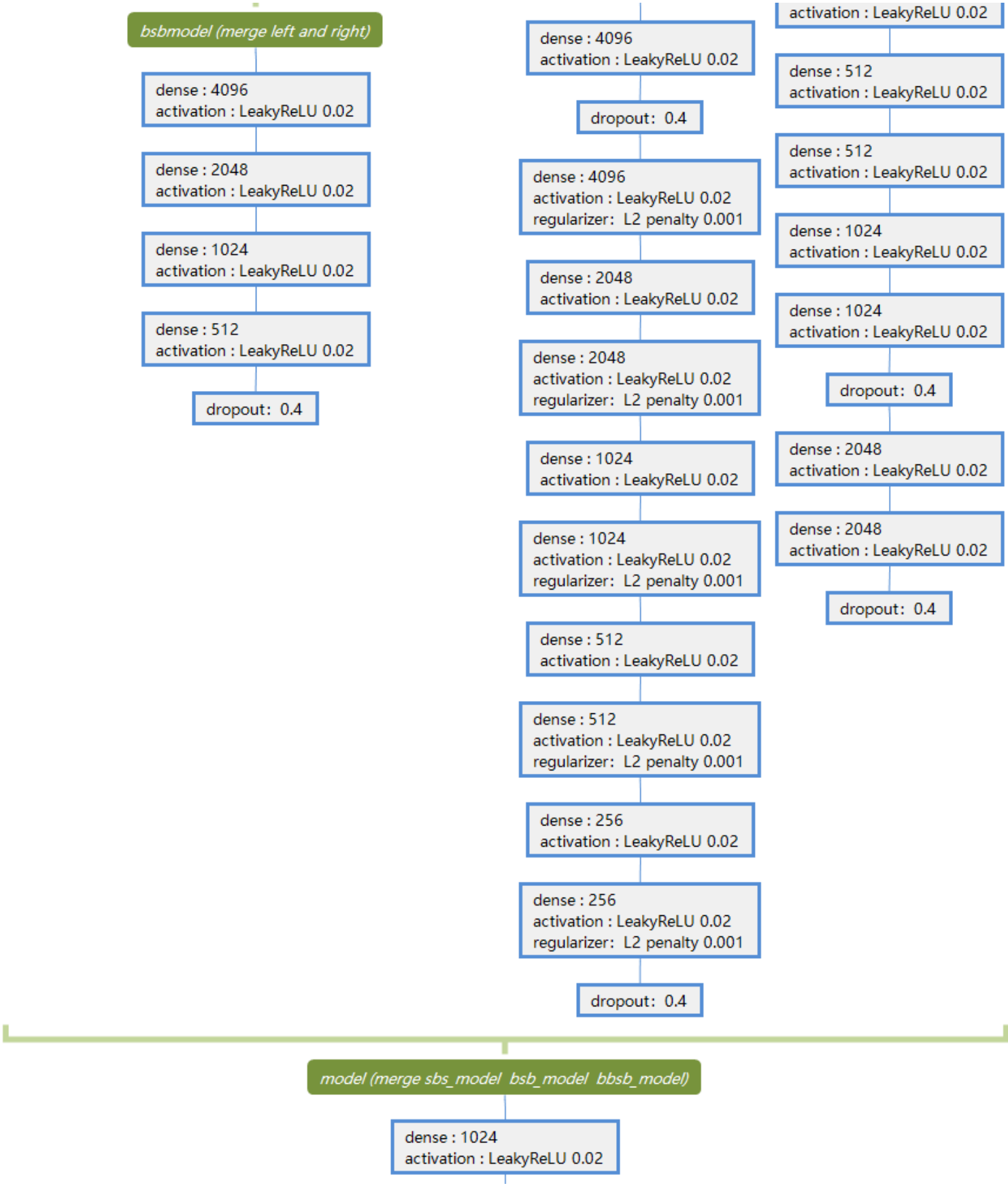
2. 深度神經網路模型介紹：

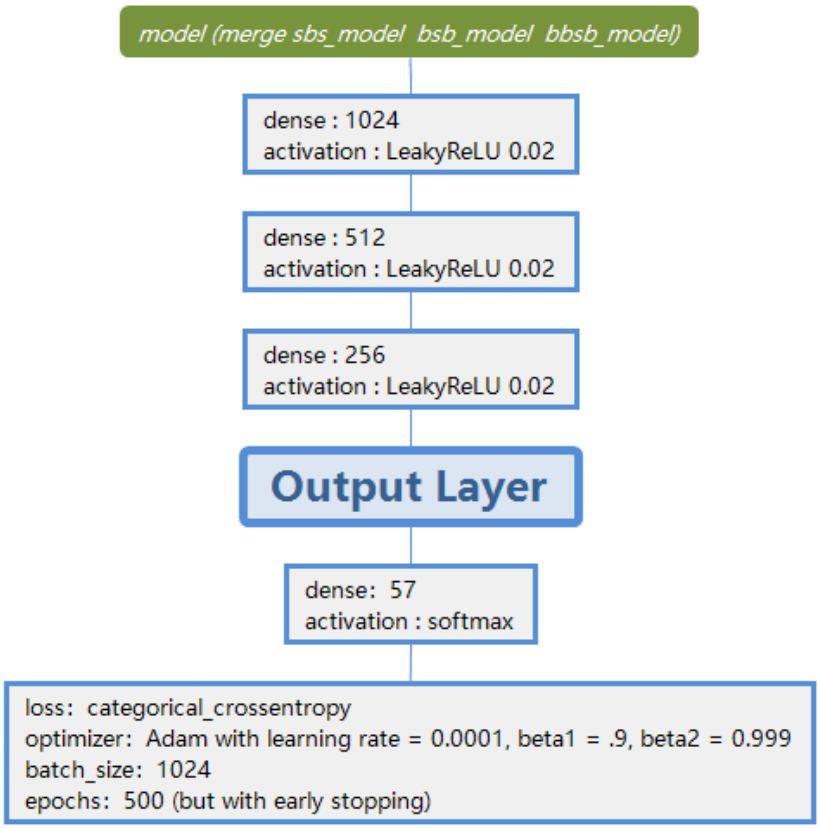
   將資料前處理過後之資料盡數放入深度神經網路模型之中。

   深度神經網路模型主要由輸入層（Input Layer)，隱藏層（Hidden Layer)，輸出層（Output Layer）所組成。我們設計模型的主要概念為：希望透過數個表現良好的小型 DNN 模型，各別抓出相異的特徵之後透過並聯的方式結合各模型的結果，最後再接上數個全連接層來達到特徵共用的目的並輸出結果。

   其模型組成圖如下：


Input Layer
left
right
sbsmodel
bbsbmodel
dense : 4096
activation : LeakyReLU 0.02
dense : 2048
activation : LeakyReLU 0.02
dense : 1024
activation : LeakyReLU 0.02
dense : 512
activation : LeakyReLU 0.02
dense : 256
activation : LeakyReLU 0.02
dense : 128
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dropout: 0.4
bsbmodel (merge left and right)


bsbmodel (merge left and right)
dense : 4096
activation : LeakyReLU 0.02
dense : 2048
activation : LeakyReLU 0.02
dense : 1024
activation : LeakyReLU 0.02
dense : 512
activation : LeakyReLU 0.02
dropout: 0.4
dense : 4096
activation : LeakyReLU 0.02
dropout: 0.4
dense : 4096
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dense : 2048
activation : LeakyReLU 0.02
dense : 2048
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dense : 1024
activation : LeakyReLU 0.02
dense : 1024
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dense : 512
activation : LeakyReLU 0.02
dense : 512
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dense : 256
activation : LeakyReLU 0.02
dense : 256
activation : LeakyReLU 0.02
regularizer: L2 penalty 0.001
dropout: 0.4
activation : LeakyReLU 0.02
dense : 512
activation : LeakyReLU 0.02
dense : 512
activation : LeakyReLU 0.02
dense : 1024
activation : LeakyReLU 0.02
dense : 1024
activation : LeakyReLU 0.02
dropout: 0.4
dense : 2048
activation : LeakyReLU 0.02
dense : 2048
activation : LeakyReLU 0.02
dropout: 0.4
model (merge sbs_model bsb_model bbsb_model)
dense : 1024
activation : LeakyReLU 0.02



3. 隨機森林模型介紹:

   將資料前處理過後之資料盡數放入隨機森林模型之中，其中的解釋變數 loc_x，loc_y 未取絕對值。隨機森林是一個包含多個決策樹的分類器，並且其輸出的類別是由個別樹輸出的類別的眾數而定。

參數	n_estimator	min_samples_split
數值	500	50

4. 邏輯式迴歸模型介紹:

   將資料前處理過後之 lat，lon，loc_x，loc_y，season，shot_distance，shot_made_flag，opponent 等解釋變數放入邏輯式迴歸模型之中，邏輯式迴歸模型將反應變數視作多項分配，並以解釋變數對其分配參數進行預測。

二、分析(Analysis)

1. 深度神經網路模型參數校調：

a. LeakyReLu = 0.02：

此參數為各層之激活函數，此處採用 LeakyReLu 理由為，希望對每一個輸入都有符合其梯度的回傳值，又，不採用 ReLu 的理由為，希望對小於 0 的輸入也給予一個相對應梯度的回傳值，而不是直接令為 0，避免失去某些重要資訊。0.02 則是多次嘗試後最佳的結果。

b. kernel_regularizer = l2(0.001)：

此參數用於調整該隱藏層中神經元之權重（weight)，將其加上一懲罰項，使得整體 loss 梯度下降較為平滑，可在模型發生過度擬和(Overfitting)問題時使用，而又由於資料量並不足夠，使用 dropout 可能會造成訓練效果大幅降低，故我們多處使用 regularizer，使用 0.001 則是希望梯度下降速度較為平緩，避免該層神經元之權重被過度輕視。

c. Dropout = 0.4：

當我們 DNN 架構越來越大時，模型將會抓處越來越多的訓練資料中的特徵，也使得過度擬和越來越嚴重，此時除了 regularizer，我們另外在某些地方加入 dropout，用此函數以加強該模型的訓練強度，0.4 意思為每次隨機斷開總連接次數之 4 成連結，而 0.4 為嘗試後的最佳結果

d. Adam, learning rate = 0.0001：

此參數用來調整 adam 之學習速率，我們在調整此參數時發現，學習速率過高時，容易使得此模型無法跳脫某些局部低點，導致每次訓練都得到相同結果，故我們設定 0.0001 來避免此局部低點。

e. Adam, beta_1 = 0.9, beta_2 = 0.999：

此參數用以調整 adam 之動量，但由於 Keras 的手冊中建議維持預設值，且十分不建議更改此二參數，故此處並無多加調整。

f. batch_size = 1024：

此參數用以調整每次放入各 batch 中的資料量，其值越高每次模型可同時學習的資料就越多，相對的也有辦法抓出更細微的特徵，但礙於硬體限制，越高的 batch_size 就需要更龐大的記憶體去暫存資料，故此處我們設定為 1024

g. 深度神經網路模型架構：

我們的模型主要由三種不同的架構組成，分別為：

i. 小大小模型（sbs_model）

ii. 大小大模型（bsb_model）

iii. 較大的大小大模型（bbsb_model）

sbs_model 的設計理念為，先透過較少神經元的隱藏層，抓出大略的資料特徵，再透過越來越多神經元的隱藏層，漸漸地將各特徵分出細節，最後再利用越來越少神經元的隱藏層來濃縮出主要的特徵。

bsb_model 的設計理念為，先建出兩個由大至小再到大的 DNN 模型（left、right)希望可以先透過大量的神經元先抓出原資料大量特徵，而後濃縮之後，再慢慢將以濃縮的特徵透過越來越多神經元的隱藏層加以放大，使特徵模糊化，進而提取出更為重要的特徵。而後合併兩個模型，再逐步收斂神經元大小來濃縮結果。

bbsb_model 的設計理念同 bsb_model，唯將其結構增加，但去除最後的合併以及濃縮特徵。

2. 隨機森林模型參數校調：

我們首先將訓練資料切成十份並以其中九份作為訓練資料集，剩餘一份作為測試資料集，並以測試資料集平均預測準確度為依據進行參數校條，調整參數包含:n_estimators，max_features，min_samples_split，min_samples_leaf。

n_estimators 代表最大迭代次數，一般來說 n_estimators 太小，容易欠擬合，n_estimators 太大，計算量會太大，並且 n_estimators 到一定的數量後，再增大 n_estimators 獲得的模型提升會很小。

max_features 代表決策樹劃分時所考慮的特徵數。

min_samples_split 限制了子樹繼續劃分的條件，若某節點樣本數少於 min_samples_split，則不會繼續再嘗試選擇最優特徵來進行劃分，可一定程度上避免過度擬和。

min_samples_leaf 限制了葉子節點最少的樣本數，若某葉子節點數目小於樣本數，則會與兄弟節點一起被剪枝，可一定程度上避免過度擬和。

最終模型選擇如下表:

參數	n_estimator	max_features	min_samples_split	min_samples_leaf
數值	500	default	50	default

3. 邏輯式回歸模型選擇：

我們首先將訓練資料切成十份並以其中九份作為訓練資料集，剩餘一份作為測試資料集，並以測試資料集平均預測準確度為依據進行向後選取法（Backward selection），以決定置入模型內之解釋變數，最終放入模型之變數包含: lat，lon，loc_x，loc_y，season，shot_distance，shot_made_flag，opponent。

4. 最終模型合成:

我們以上述三種模型對測試資料的 action_type 進行預測，並使用投票的方式，從三個模型預測結果中採用多數決以此決定最終預測結果。若三者意見相左以邏輯式迴歸模型預測結果為基準。最終預測準確度為 0.7236。