**Competition1**

指導教授：李政德　教授

組員:王子豪H24031346 周逸平H24036087

1. **模型介紹(Description)**

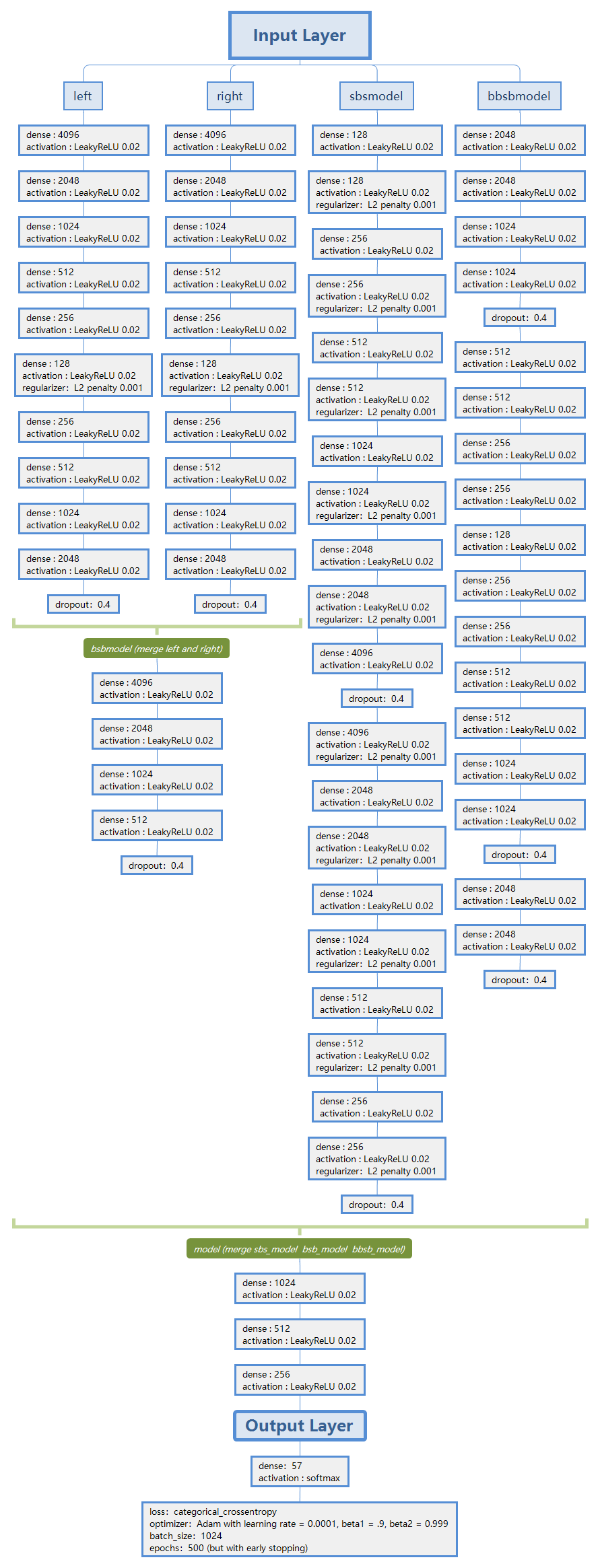
我們分別使用了隨機森林模型（Random Forest Model）、邏輯式迴歸模型（Logistic Regression Model）、深度神經網路模型（Deep Neural Network Model）對Kobe Bryant的投籃類型進行預測，之後使用投票的方式，從三個模型預測結果中採用多數決以此決定最終預測結果。若三者意見相左以邏輯式迴歸模型預測結果為基準。

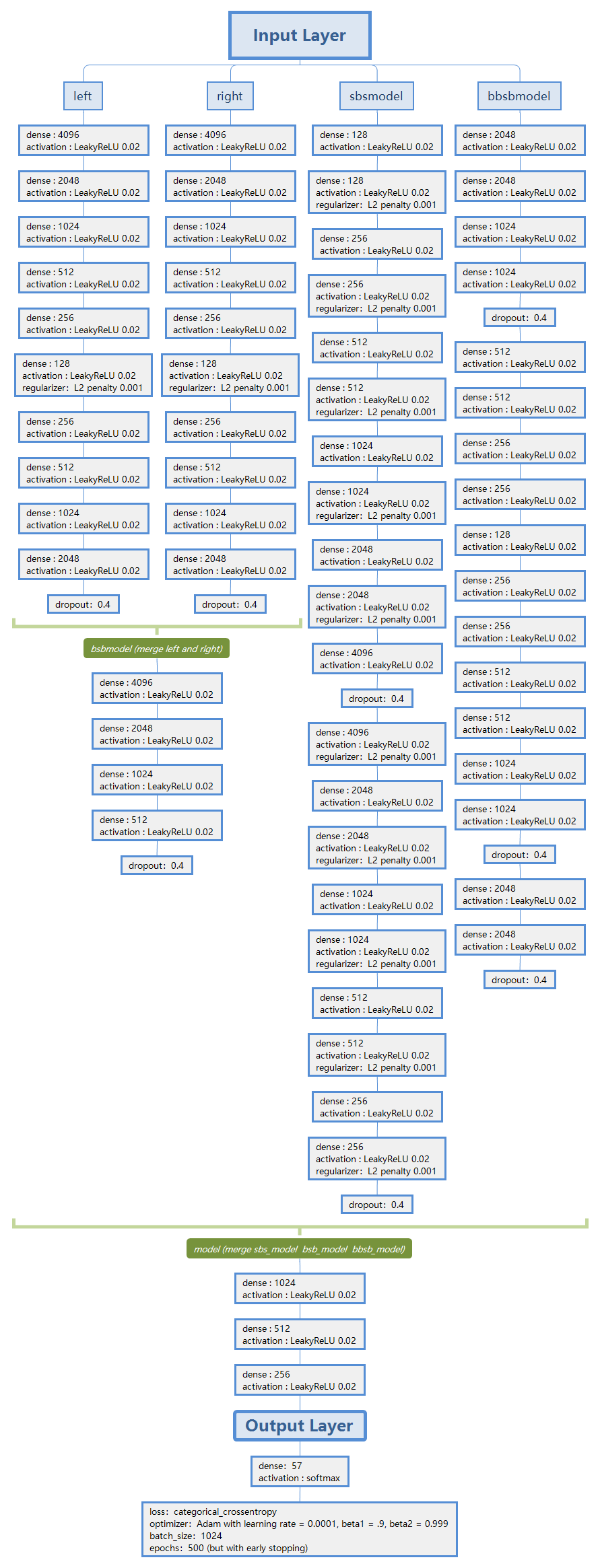
1. 資料前處理：
2. 將game\_date變數轉換為浮點數型態，以此將其視作連續型變數。
3. 將loc\_x，loc\_y此二變數取絕對值，此外將playoffs，period，season，shot\_made\_flag，shot\_zone\_area，opponent，action\_type等變數視作類別變數並轉換成虛擬變數（Dummy Variable）。
4. 深度神經網路模型介紹：

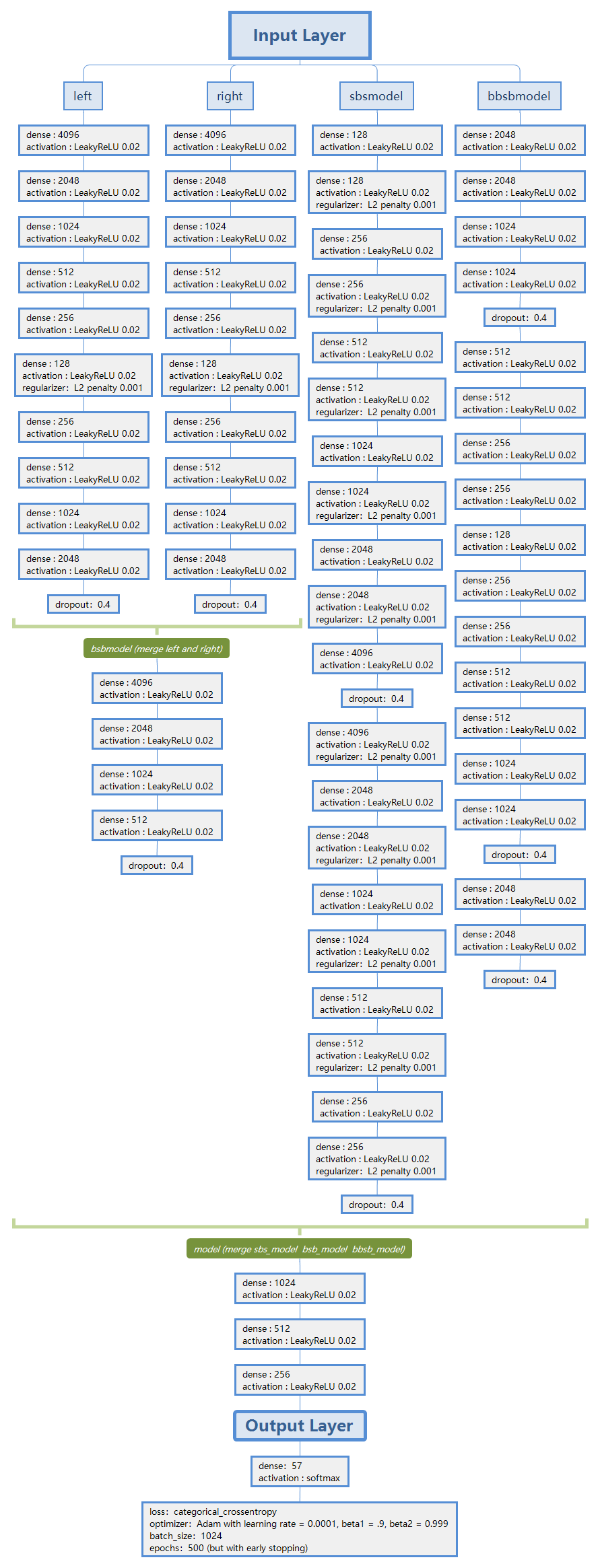
將資料前處理過後之資料盡數放入深度神經網路模型之中。

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：







1. 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](https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91)的[分類器](https://zh.wikipedia.org/wiki/%E5%88%86%E7%B1%BB%E9%97%AE%E9%A2%98)，並且其輸出的類別是由個別樹輸出的類別的[眾數](https://zh.wikipedia.org/wiki/%E7%9C%BE%E6%95%B8)而定。

|  |  |  |
| --- | --- | --- |
| 參數 | n\_estimator | min\_samples\_split |
| 數值 | 500 | 50 |

1. 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：

1. 小大小模型 (sbs\_model)
2. 大小大模型 (bsb\_model)
3. 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。