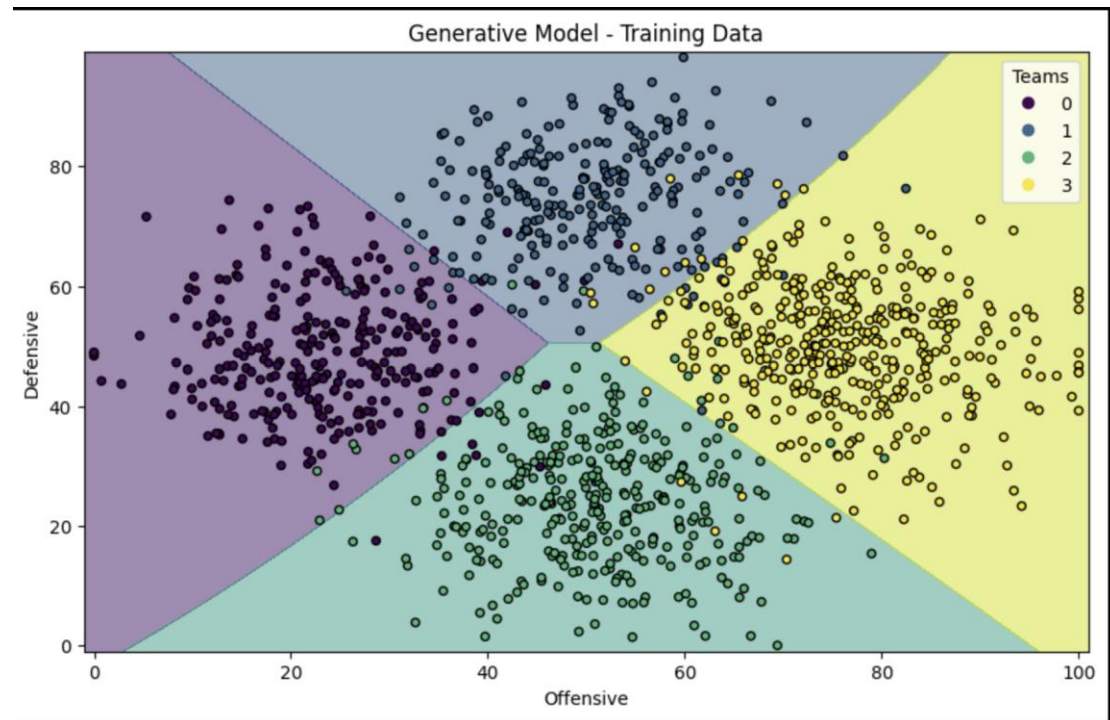
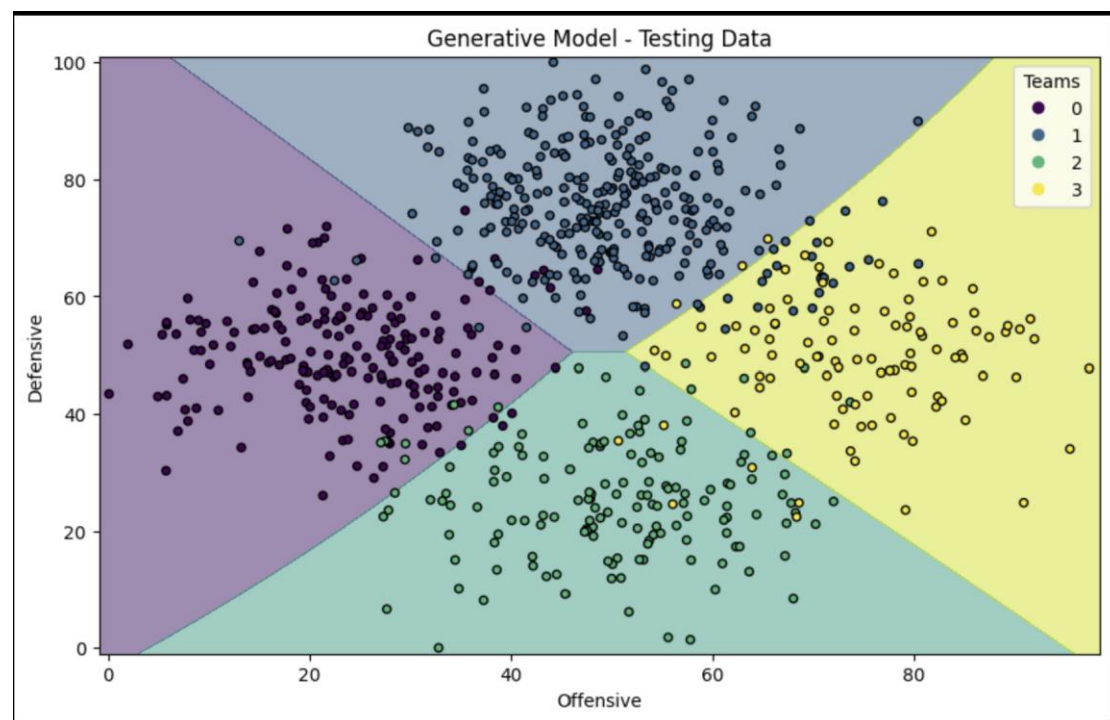


機器學習 HW02

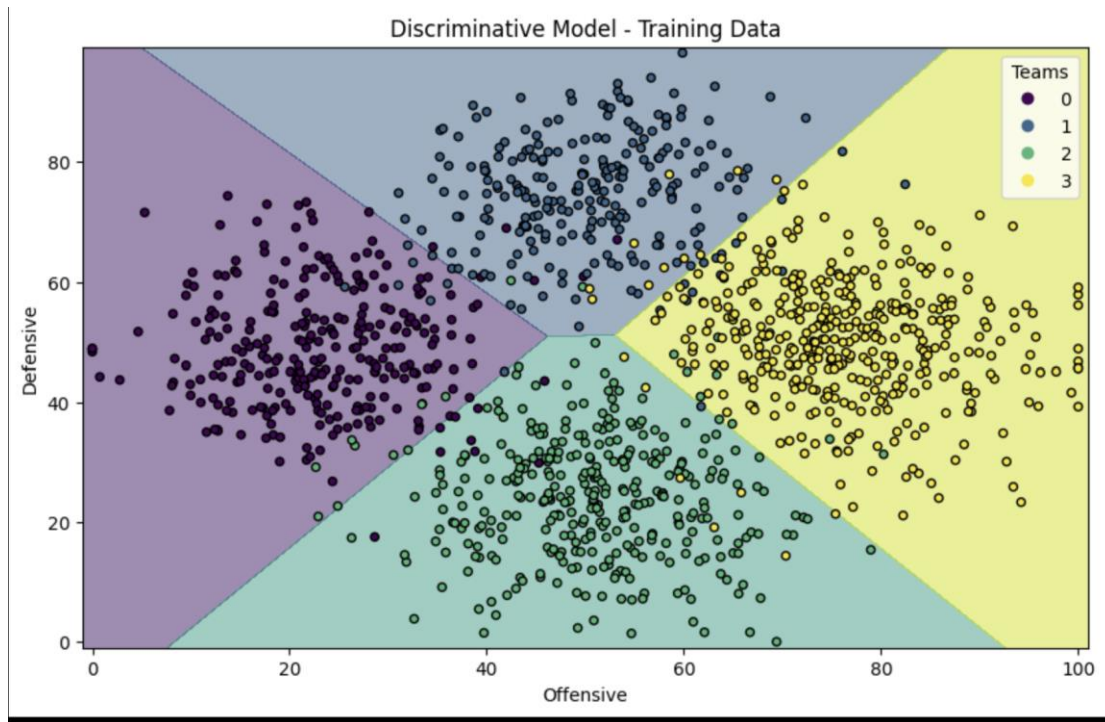
PART1 圖片:



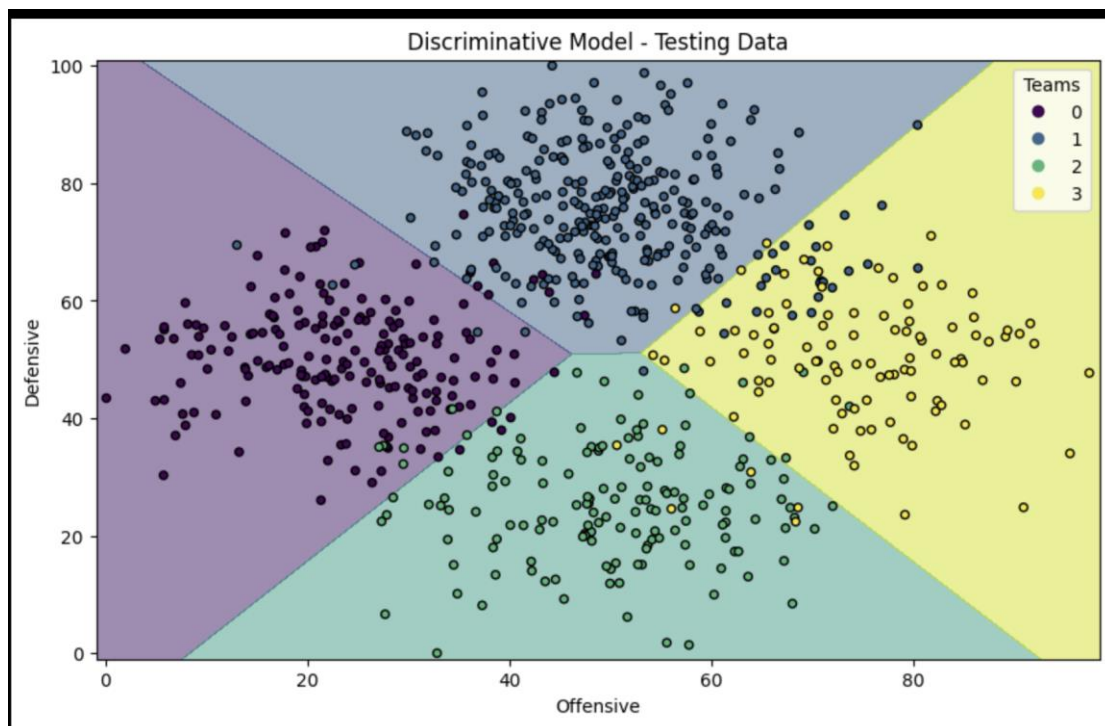
Accuracy=0.943077



Accuracy 0.906667



Accuracy= 0.942308

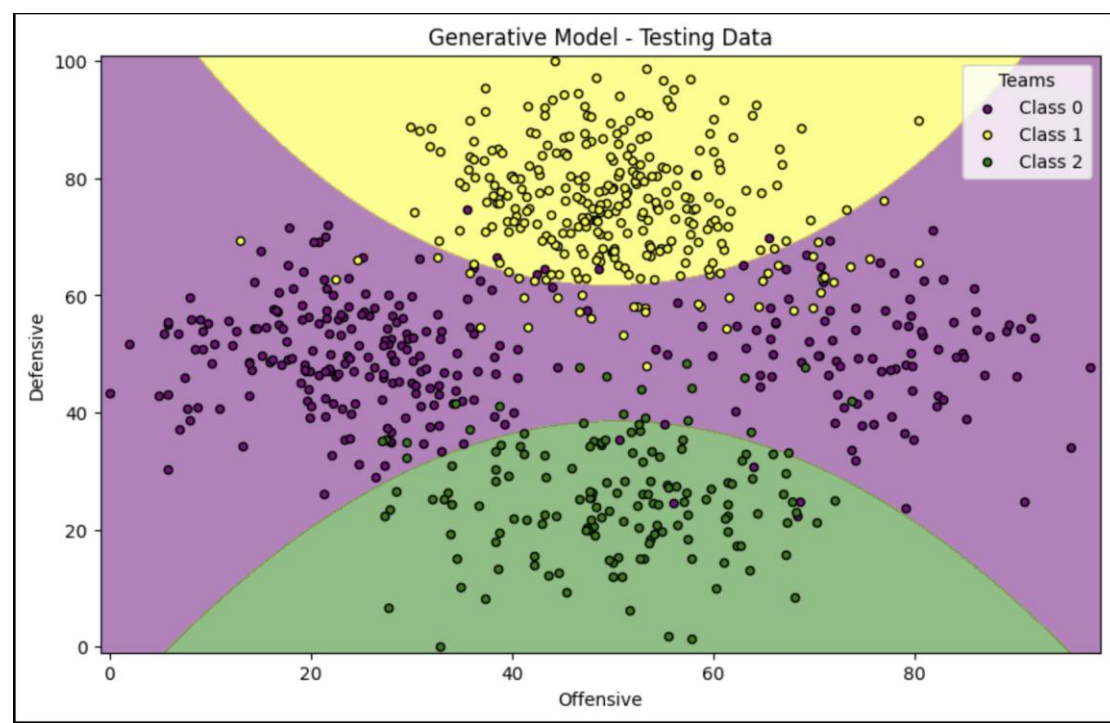


Accuracy =0.909333

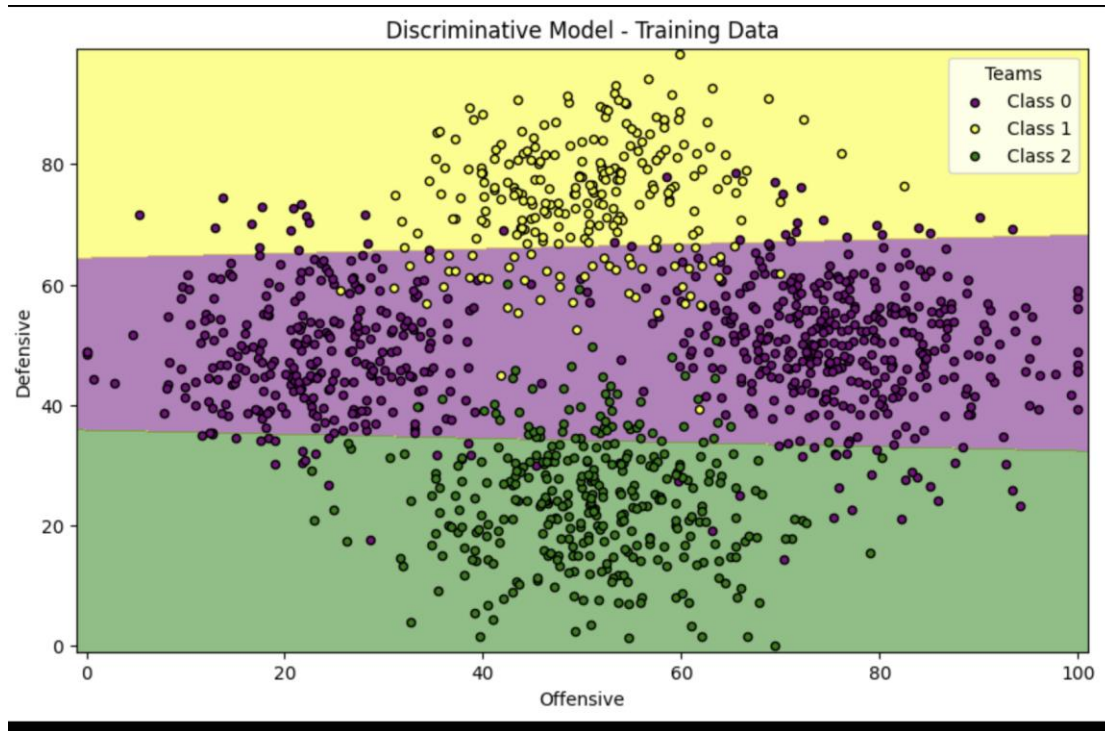
Part2 圖片:



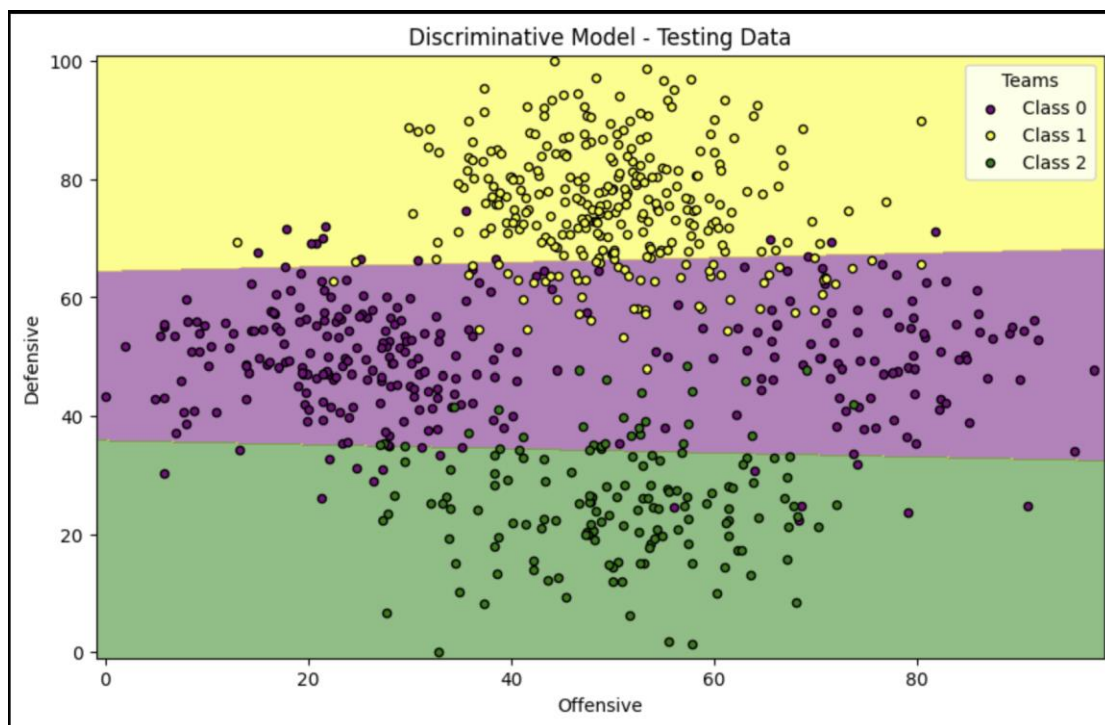
Accuracy= 0.924615



Accuracy= 0.897333



Accuracy= 0.853077



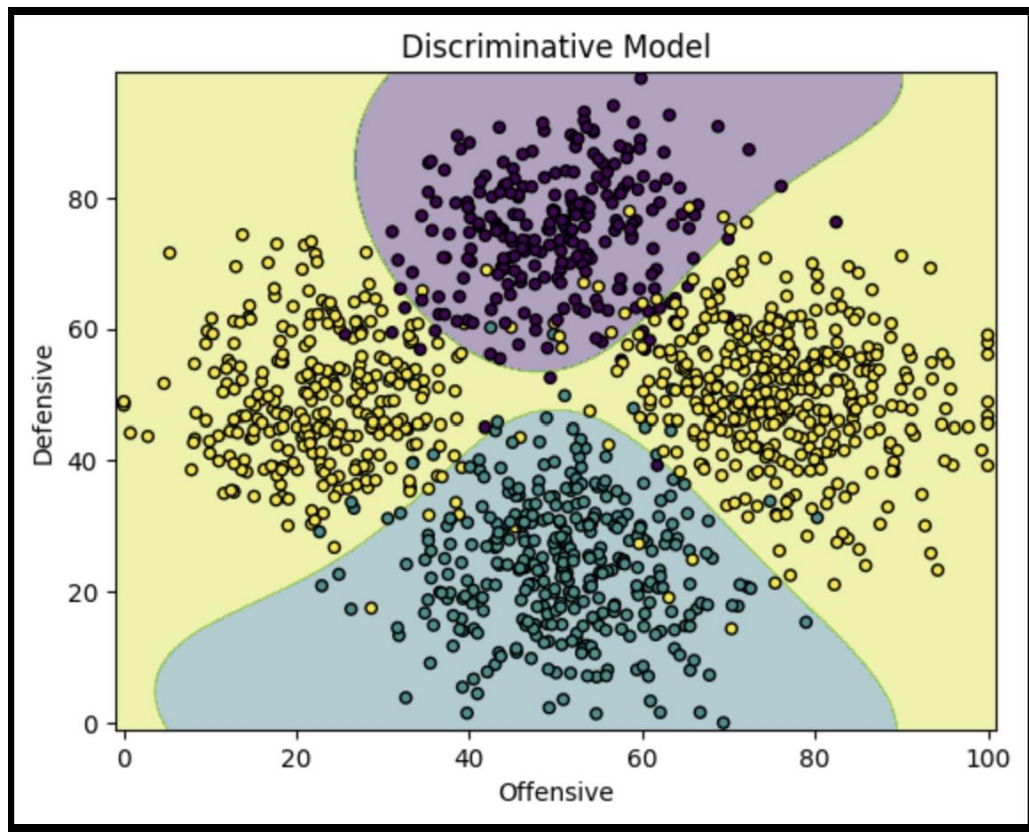
Accuracy= 0.842667

觀察與討論

生成模型與判別模型的差異

- 生成模型:
 -
 - 學習數據的聯合概率 $P(X,Y)P(X,Y)$ ，捕捉數據分佈。
 - 比較不會有過擬合的問題。
 - 由於生成模型需要生成 Y 的分佈函數，這個分佈函數可能會受到一些異常點的影響，變得不那麼準確。生成模型在學習數據的聯合概率分佈 $P(X,Y)P(X,Y)$ 時，會嘗試理解整個數據集的特徵與標籤之間的關係，這使它們對於異常值或是噪聲數據特別敏感。如果數據中包含異常點，這些點可能會對模型學習的整體數據分佈產生不成比例的影響，從而導致模型的預測性能下降。
- 判別模型:
 - 估計條件概率 $P(Y | X)P(Y | X)$ ，關注類別之間的邊界。
 - 如圖所示，為橫切面且未包含所有數據點導致準確率不夠準。
 - 在邊緣區分沒那麼明顯的數據表現較差

推測可能是我使用 **logistic regression** 線性模型影響，我另外使用 **svf** 非線性模型來做判別模型則會表現較好，如圖所示:



得到的準確度稍微提升，且圖中也可明顯看出有邊界明顯分類出不同隊伍。

Training Accuracy: 0.9453846153846154

Testing Accuracy: 0.9106666666666666

以上為寄信與助教討論過的內容，因此在報告中呈現出來。

Code 實現方法:

生成模型

生成模型通過學習數據的聯合概率分布 $P(X,Y)$ 來進行預測。在這個例子中，使用了朴素貝葉斯分類器，這是一種簡單且有效的生成模型，它假設各特徵間條件獨立。

1. 模型創建與訓練：

- 使用 **GaussianNB()** 來創建一個朴素貝葉斯模型的實例。
- 調用 **fit()** 方法來訓練模型，使用訓練數據 **X_train**（攻擊和防守特徵）和 **y_train**（隊伍標籤）。
-

2. 進行預測：

- 使用 **predict()** 方法對訓練數據和測試數據進行預測，得到 **y_pred_train_gnb** 和 **y_pred_test_gnb**。

3. 性能評估：

- 計算模型在訓練集和測試集上的準確率，並生成混淆矩陣，以評估模型的性能。

判別模型（邏輯回歸）

判別模型直接學習輸入 XX 到輸出 YY 的映射關係 $P(Y | X)$ ，專注於決定邊界。

1. 模型創建與訓練：

- 使用 **LogisticRegression()** 來創建一個邏輯回歸模型的實例，指定 **multi_class='multinomial'** 以支持多類分類，並使用 **lbfgs** 求解器。
- 通過 **fit()** 方法訓練模型，同樣使用 **X_train** 和 **y_train**。

2. 進行預測：

- 使用 **predict()** 方法對訓練和測試數據進行預測，得到 **y_pred_train_lr** 和 **y_pred_test_lr**。

3. 性能評估：

- 計算模型在訓練集和測試集上的準確率，並生成混淆矩陣。

繪制決策邊界

代碼中還包含了一個繪制決策邊界的功能，這有助於視覺化每種模型如何在特徵空間中區分不同的隊伍。此功能通過生成一個格點，並在格點上計算模型預測的類別，然後將這些類別顏色編碼顯示出來，最後加上原始數據點的散點圖。