

Classifier Report

I. Environment

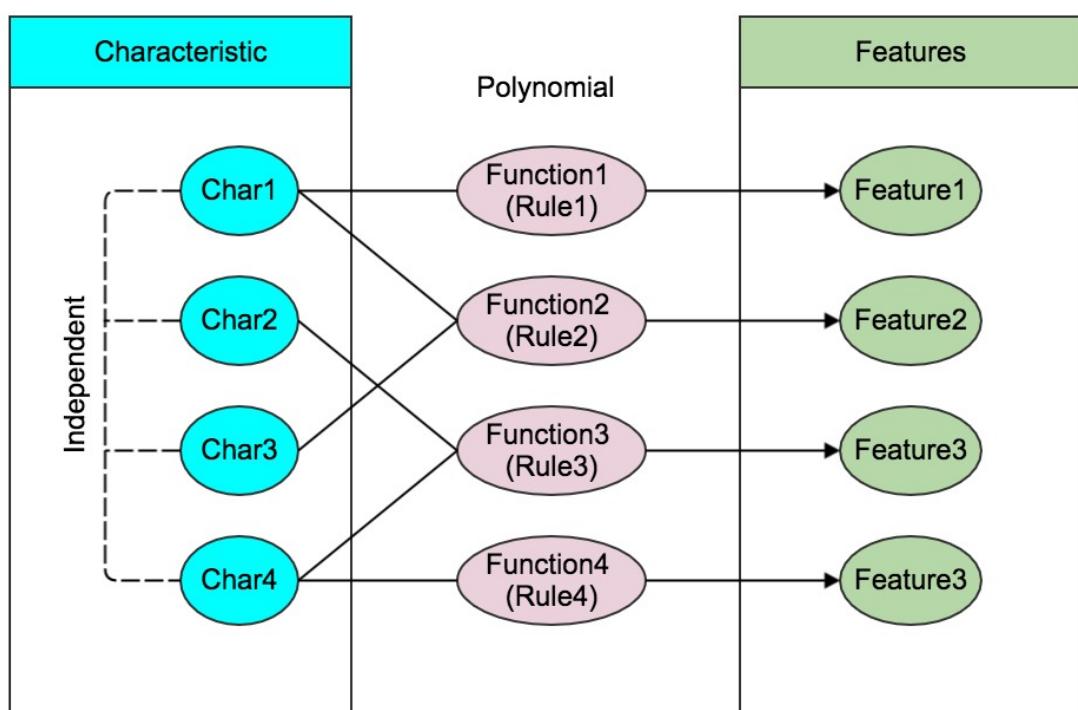
OS: Ubuntu 16.04 KVM

Python version: 2.7.12

Library: Numpy, SciPy, sklearn

II. Data Generator

➤ How to generate features



圖一、Features 生成示意圖

為了資料的彈性，我假設可觀察到的 Feature 皆由獨立的 Characteristic 搭配 Polynomial 的 Function 所生成，此 Function 即為作業要求中的 Absolute Rules，Features 與 Characteristic 的差異在於是否獨立，Characteristic 可以想成世上最細節的元素，依循絕對的法則後產生的現象即為我們所看到的 Features，此概念啟發於拉普拉斯妖 (Démon de Laplace)。在 `datagen.py` 中，允許可以透過參數調整 Characteristic 數目、Features 數目、線性與否等參數。

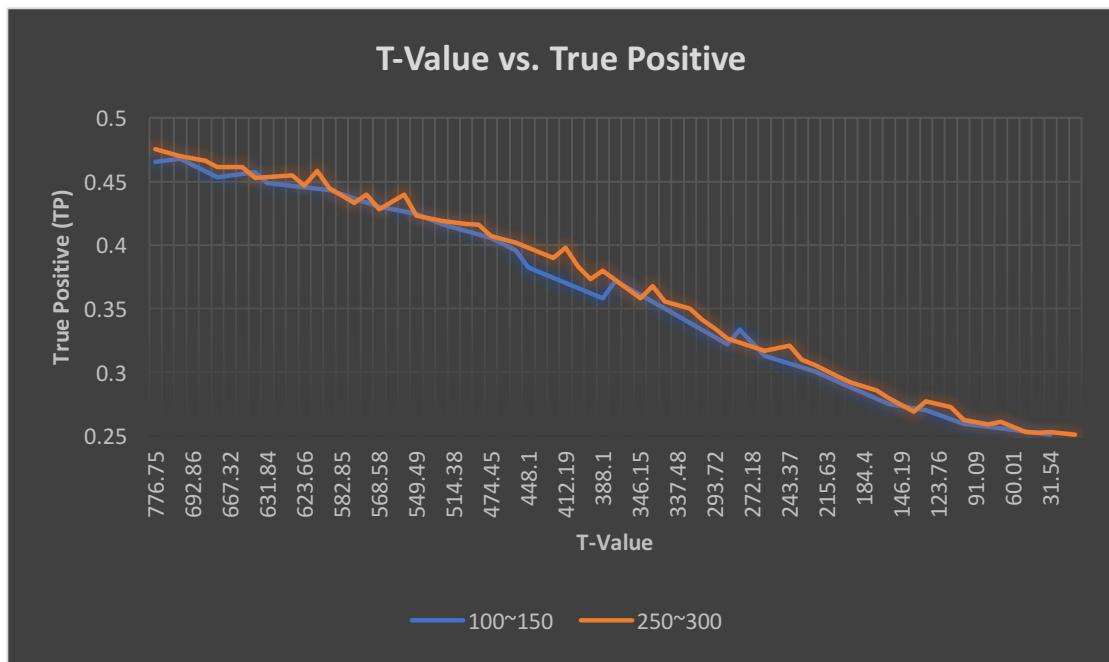
III. Experiments

以下實驗結果保存 result.txt 中，實驗所使用的數據保存在 npz 資料中。

➤ Confusion Matrix

以下實驗中的 Characteristics 皆符合 Normal Distribution，為了方便計算 T-values，所有 Characteristics 的 mean & variance 皆相同，如此兩分類中的各 Characteristic 之 T-value 皆相等，將資料的 75% 作為 Training、25% 作為 Testing，以 Decision Tree 分類時所產生的 Confusion Matrix 具有對稱性，即 TP & TF 相近，NP & NF 相近，故以下僅觀察 TP 值作為效能表現的指標。

➤ T-value vs. True Positive

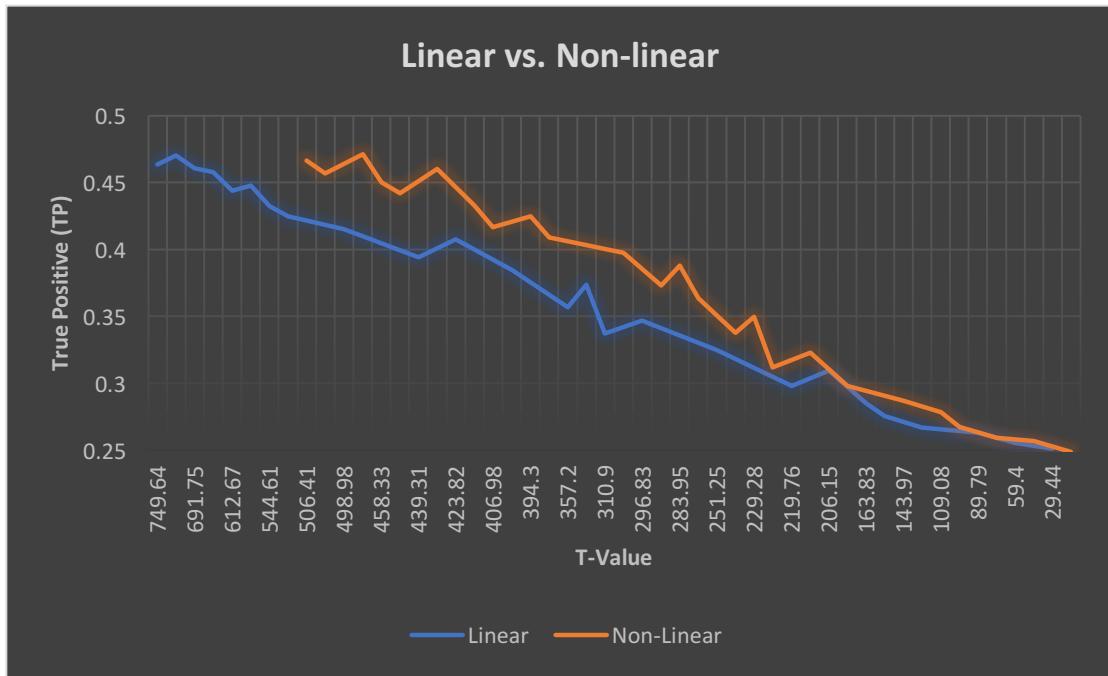


圖二、T-Value vs. True Positive

使用 2 個 Characteristics 以線性組合的方式產生 2 個 Features，總共兩組數據且每組數據中有 A 與 B 兩種分類，第一組數據 A 的 Characteristic 的 mean 將會從 100 逐漸成長到 150，而 B 的 mean 將會固定為 150，第二組數據的 A 的 mean 由 250 成長到 300，B 的 mean 則固定為 300，以此方式搭配 Decision Tree 分類並觀察 T-value 與 True Positive 的關係。

由上圖可知，不論 mean 為 150 或 300，True Positive 的機率與 T-value 直接相關，在 Student t-test 中，T-value 代表兩個 Characteristic 的差異程度，當 T-value 越大時，則 Features 的的差異程度也越大，因此當 T-value 逐漸變小時，代表兩分類的差異性逐漸縮小，使分類器的 TP 越來越小。

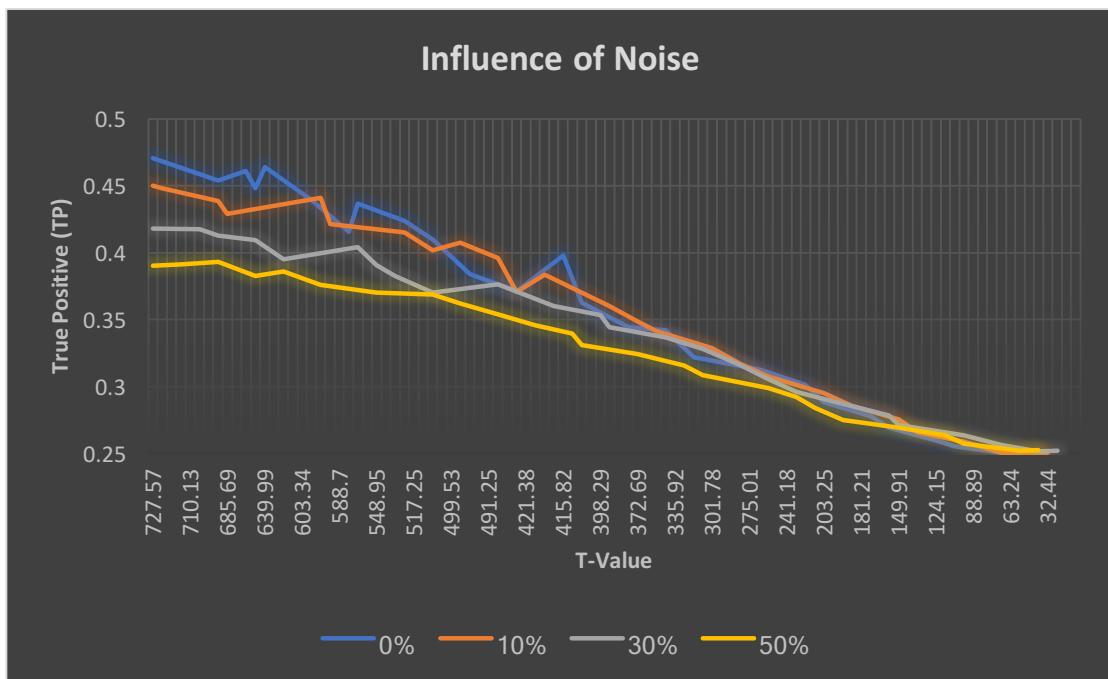
➤ Linear vs. Non-Linear



圖三、Linear & Non-Linear

使用 2 個 Characteristics (mean=100~150, variance=20) 以 Linear 及 Non-linear (Polynomial, power=2~4) 的方式組成 2 個 Features，使用 Decision Tree 觀察其結果差異，如圖三所示，因為 Decision Tree 為 Non-linear 分類器，因此可以良好的進行分類，Non-linear 結果甚至比 Linear 好。

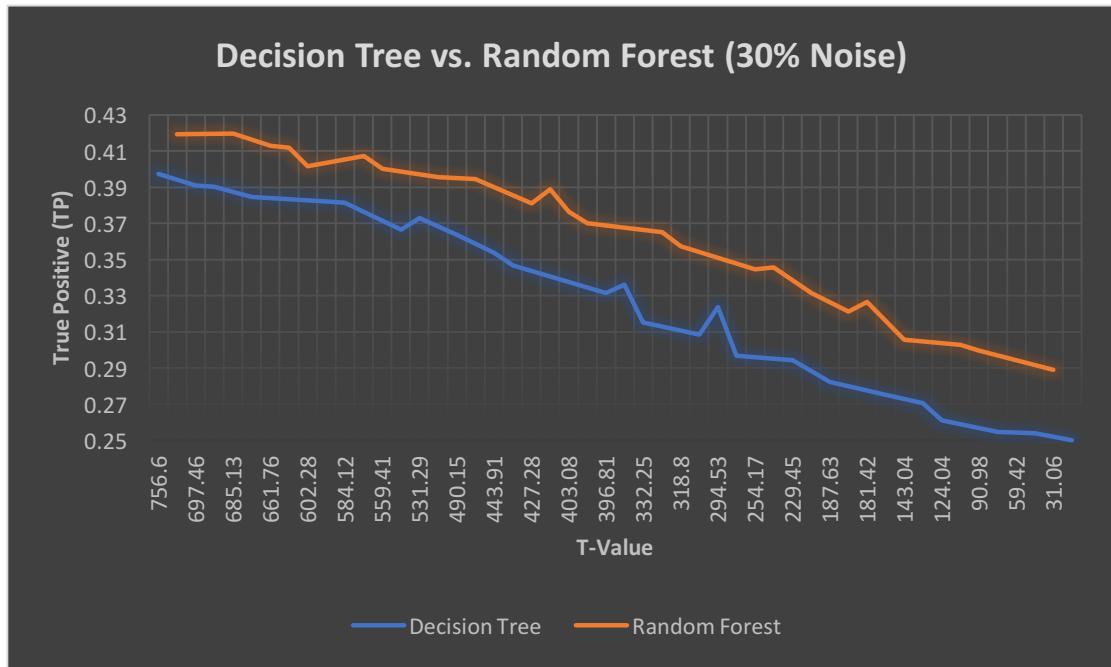
➤ Influence of Noise



圖四、Influence of Noise

雖然 Decision Tree 可以在 Linear 及 Non-linear 表現都不錯，但在有 Noise 的干擾下表現會迅速下降，為了生成 Noise，本實驗先將兩分類的資料混和，再找出各個 Feature 的最大許最小值，從 $0.5 * \text{最小值} \sim 1.5 * \text{最大值}$ 中 Random 生成隨機資料，而分類結果亦是隨機的，如圖四所示，四條線的百分比關係到有多少 Noise 生成，如 50% 則代表 Noise 數目為 $0.5 * \text{原始資料數目}$ ，如圖四所示，當 Noise 比例上升時，Decision Tree 的分類表現 (TP) 迅速下降，可以 Decision Tree 容易因為 Noise 而產生 Overfitting 的現象。

➤ Decision Tree vs. Random Forest (30% Noise)



圖五、Decision Tree vs. Random Forest (30% Noise)

為了減緩 Decision Tree 容易受到 Noise 而 Overfitting 的現象，本實驗比較 Random Forest 是否能在 Noise 存在時有較好的表現，延續前一個實驗，添加 30% Noise 之後分別執行 Decision Tree 及 Random Forest 兩個分類演算法，結果如圖五所示，Random Forest 在整體上比 Decision Tree 有更高的 TP 比率，在 $T\text{-value}=31.06$ 時，Decision Tree 的 TP 值已經落入接近 25%，意即與兩者隨機猜測無異，然而 Random Forest 却能維持在 30% 左右的準確率。