PreProcessing

-demographics:

Introduction:

此表共有6163列9欄 ，紀錄顧客的人口統計數據，包含「性別」，「年齡」，「是否小於30歲」，「是否為年長者」，「結婚與否」，「是否有同居人」，「同居人數」等。其中欄位之間存在相依關係，例如「年齡」，「是否小於30歲」，「是否為年長者」等。

因此資料補缺失值方式採下列邏輯：

* 「性別」遇缺值則補眾數;
* 「年齡」遇缺值則檢查「是否小於30歲」，如為是，則補30歲以下資料平均年齡，如為否，則接續檢查「是否為年長者」，如為是，則補65歲以上資料平均年齡，如為否，則檢查「是否結婚」，如為是，則補30歲以上且65歲以下結婚人口之平均年齡，如為否則補30歲以上且65歲以下未結婚人口平均年齡。
* 「是否小於30歲」遇缺值則根據已補好的年齡數判斷。
* 「是否為年長者」遇缺值則根據已補好的年齡數判斷。
* 「結婚與否」遇缺值則判斷該缺值對應的「年齡」是否大於已接婚人口均年齡加上為結婚人口均年齡之平均。如是，填入已結婚，如否則填入為結婚。
* 「是否有同居人」如遇缺值則檢查「是否結婚」，如為是則補是，如為否則補否。
* 「同居人數」如遇缺值則檢查「結婚」，如為是則補同居人數之眾數，如為否則補0

-satisfaction :

此表共有6164列2欄，「ID」及「滿意度」。補值方式採下列邏輯：

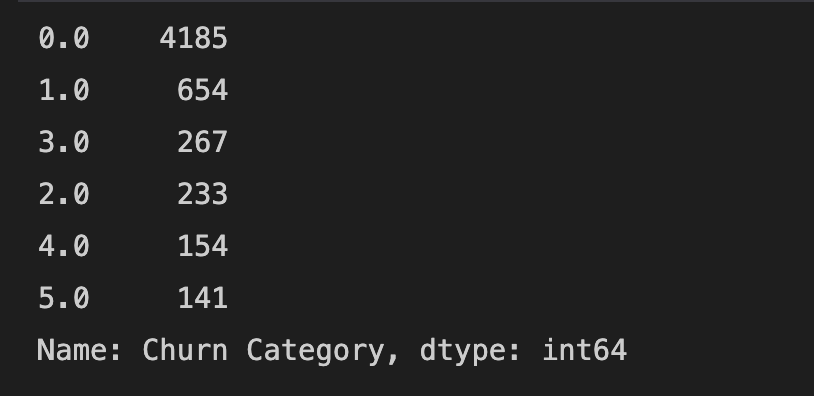
* 「滿意度」遇缺值則補滿意度之眾數。

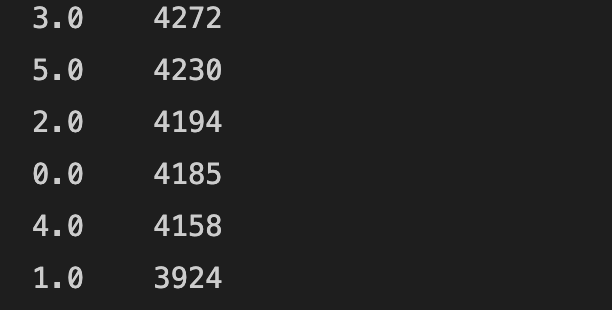
Algorithm :

SVM

使用svm訓練前，因觀察到「churn category」大部分皆為0，於是我們先對資料樣本做oversampling，目的在於使資料平均，且讓svm能學到除了0之外的分類。

Oversampling的方式以「churn category」等於0為標準，其他各類別資料分別複製其數量至接近之比例。





所選用訓練之資料集為所有表格之集合，總共41個特徵值，訓練svm模型時，kernel function採用Gaussian kernel，用意在於特徵數較多，因此映射至無線維以便求得較佳之分線，再者，為了避免C值過度擬合，C值設為0.1，gamma值則交由套件自動算出。

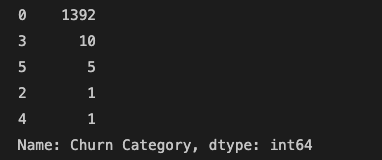
且因「churn category」為多類別之分類，分類上選擇ovo(one-versus-one)產生N\*(N-1)/2共10種分類器選擇出最適合的分類。

**結論**：

E\_in = 0.843,

E\_validate = 0.832

乍看之下似乎獲得不錯結果，Kaggle上之Fscore卻只有0.14，實際預設TestID類別如下：



顯示出訓練出之svm模型在類別0以外之分類並沒有學習的好。

Self-Defined-Decision-Tree :

此名詞是我們定義出的一套概念，底層實作方法可抽換AdaBoost或XgBoost等演算法，概念說明如下：

Introduction:

在訓練各種演算法的模型時，我們採用所有表格的欄位一起進行訓練，因此在預測TestID的類別時理當需要這些特徵欄位輸入模型中進行運算，然而每筆TestID不一定皆存在於所有表格中，代表TestID並不含所有欄位資訊，因此將會面臨TestID並無足夠特徵數輸入由所有表格欄位訓練出之模型的狀況。

Solution ：

我們採三層樹的架構以處理上述之狀況。第一層樹由所有表格的欄位訓練出之模型進行預測，如TestID有缺失任一欄位無法進行預測，則進入第二層樹，以「services」資料表進行訓練所得出的模型進行預測，如有缺失「services」表格中任一欄位資料則進入下一層樹，由「demographics」表格所訓練出之模型進行預測。用意在於，避免TestID缺乏特徵資訊而無法進行預測導致只能隨機猜測其類別或是固定猜測其類別。