機器學習與方法學之在房屋價格預測的實務應用

組員：資工 4A 林語新 資工 4A 陳政文 資工 4A 繆穩慶　指導教授：蕭子健

研究動機

上學期我們上了人工智慧和機器學習的課，為了更加了解這些演算法，並且增加一些機器學習的實作經驗。我們在Kaggle找到這個房價預測的競賽，在此專題中我們將使用Decision Tree、K Nearest Neighbor Classifier以及Naive Bayes' Classifier演算法，它們將藉由網站所給的數據來進行學習房價的預測，我們將比較各個演算法的訓練時間、預測時間以及預測準度並分析它們的差異。

現有相關研究概況及比較

機器學習包含了許多不同的方法，在這次的實驗中我們將使用大家最常用也為基本的幾種方法。

Decision Tree

是一種藉由特殊的樹來建構決策流程的方法。在Decision Tree中，每個node會以資料中變異量最大的要素進行分類，以期望做到最佳分類。Decision Tree 在實作時易於理解且能夠在相對較短的時間內做到良好的預測結果。

在Decision Tree 中，一般來說我們為了避免所產生的結果會有overfitting的狀況產生，會有所謂「剪枝」的動作。在此專題中我們運用到的是預先剪枝，也就是在一開始就決定一棵樹生長的最大深度，使預測結果能夠更具有一般性。

K Nearest Neighbor Classifier

KNN的過程，只需將所有訓練數據存儲在下來。 並將其數據結構成一個簡單的列表。 在預測階段，當模型用於對新數據進行預測時，便會計算預測資料與表中每個數據之間的要素在空間中的距離，並且模型返回的預測是目標特徵級別 在特徵空間中最接近的數據。

最近鄰模型中使用的距離方式有許多種。例如 Manhattan 距離、Euclidean 距離等。在此專題中我們將會以上述兩種方式進行分析。在KNN中，搜尋資料有許多種方式，本次專題會使用到KD-Tree的方式進行搜尋。

Naive Bayes' Classifier

Naive Bayes' Classifier 是一套由Bayes' Theorem 為基礎的監督學習演算法，在每一對特徵之間採用獨立的假設，因此Naive Bayes' Classifier 中描述要素的條件概率僅適用於目標要素。在此專題中，我們將會使用高斯分佈、白努力分佈與多項方式來進行Naive Bayes Classifier。

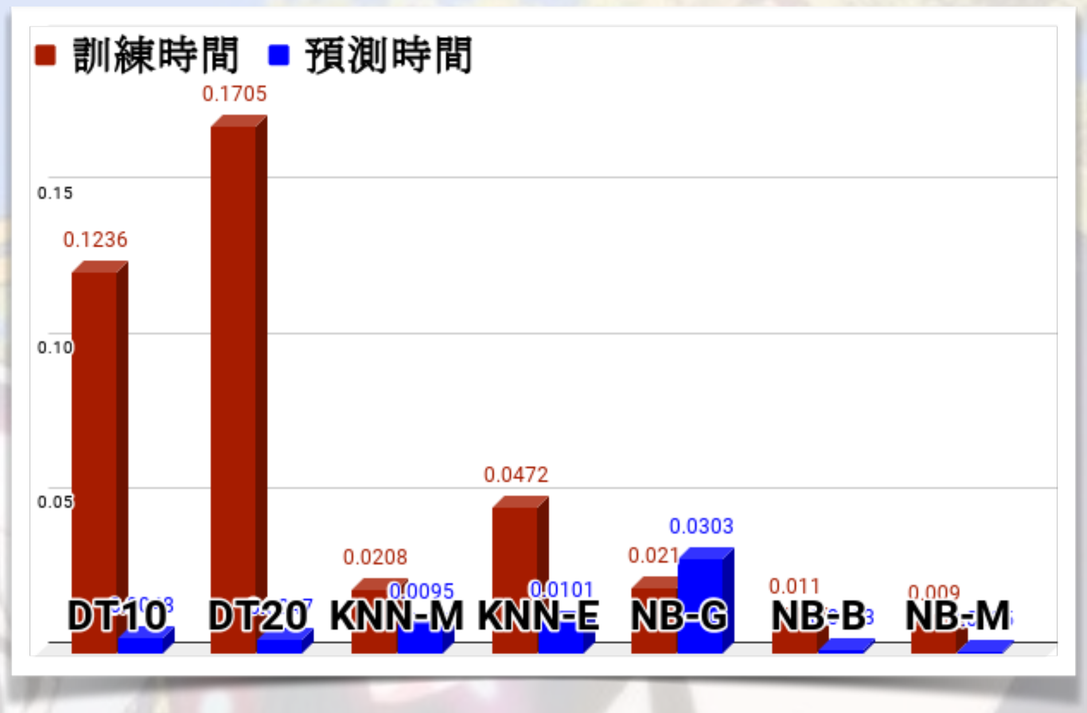
研究方式

我們將使用python中的scikit-learning套件與其他相關套件進行我們的實驗。 我們這次所使用的資料為Kaggle競賽網頁中所提供的House Prices: Advanced Regression Techniques這道題目。我們將各種不同方法（Decision Tree、KNN、Naive Bayes' Classifier等）以K-fold的方式來檢測方法之間的差異。其差異包括訓練時間、預測時間與特定範圍的預測準確度等。

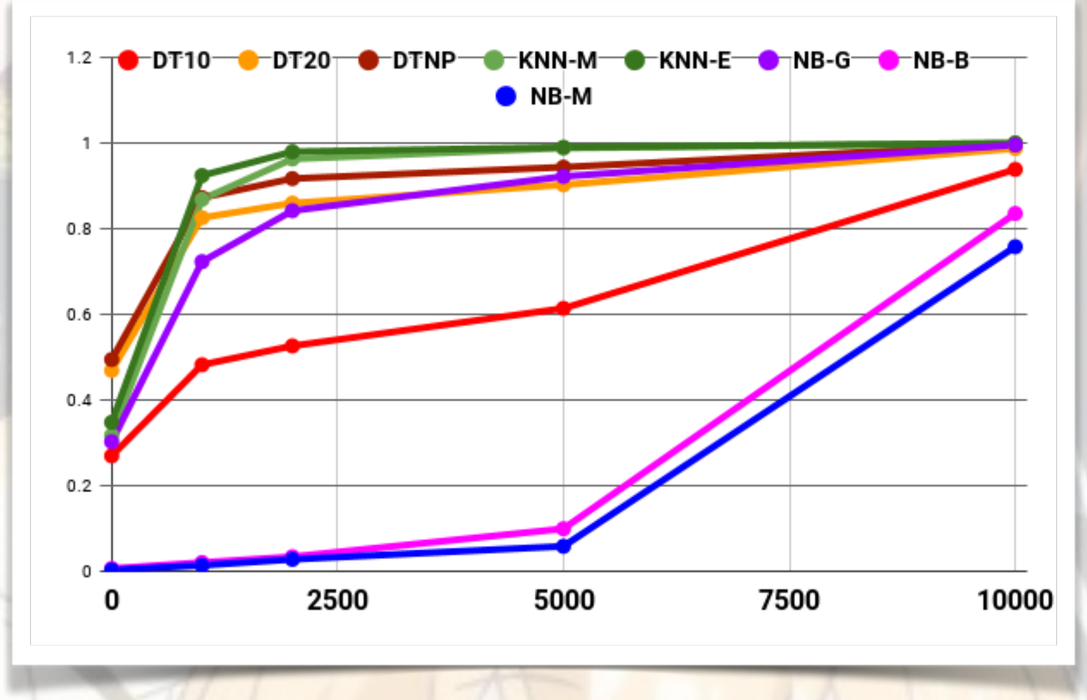
- 下載資料並處理資料裡的內容，以符合分類的規則
- 將處理完成的資料使用scikit-learn套件的機器學習方法，並以K-fold方式來驗證資料的準確程度
- 將預測結果在可容許範圍內（0，2000，5000，10000）的準確值做統計，並再與訓練時間、預測時間、準確率三者進行分析比較

研究結果

訓練時間與預測時間之比較：



不同方法在容許誤差內的準確度比較：



較多時間進行訓練的方法在測試估算時會有較快的速度。以我們這次的實驗看來資料約1500筆或許並無法凸顯出估算的速度，但要是測試資料的data一大的話，時間的差異或許便能明顯地看出了。

而關於Naive Bayes' 的算法中，白努力跟Multinomial 中又顯得很差，我們認為是因其設計為0/1的二元形式，但房價元素的內容絕非正反而已，因此也造成了效果上的低落。

當誤差在較小的情況時decision tree 的準確率看起來比較高，但誤差範圍一擴大時KNN會有較高的準確率產生。這樣的結果或許可以讓我們考慮到decision tree與KNN的差異。藉由距離的預測方式或許在極為準確地判定時不會那準確，但把誤差值擴大便能將其包圍著。相反的decision tree 以絕對的的數來切割或許能夠將最重要的東西進行區分導致誤差為0的準確度較大，但一但誤差擴大時便無法完全掌握。

結論

我們認為房價預測並不需要要求零預測誤差，因此就這個房價預測模型而言，在有少量誤差內準確度較高的K Nearest Neighbor Classifier是個比較適合的演算法，但是這並不代表其他的演算法不好，只是在這個房價預測不適合。而若以完全預測為目標的話，勢必得尋找其他演算法，畢竟，這三種演算法零誤差時的準確率不甚理想，Naive Bayes Classifier甚至趨近於零。

我們之後預計會再學習更多演算法，譬如說機器學習的其中一個分支-強化學習當中發展出來的XCS(accuracy-based Learning classfier system)這種online learning的演算法，或是GAssist這種非常仰賴Genetic Algorithm的offline learning演算法，還有像是類神經網路等方法。並且我們也會使用更多不同的dataset，期望能找出各種演算法最好的使用情況。