**機器學習導論期末分析報告書**

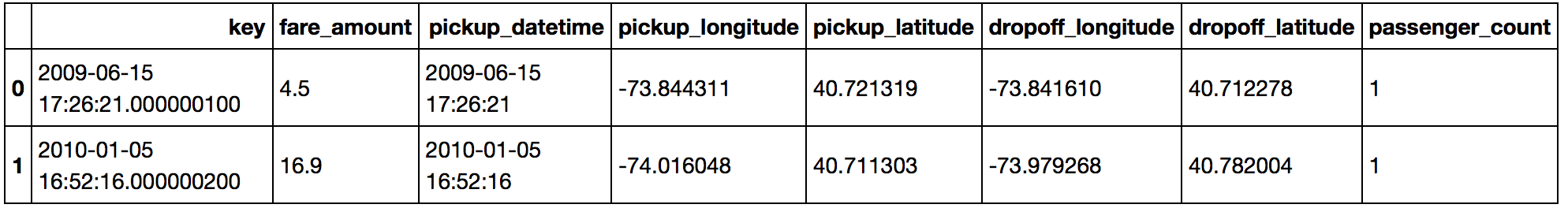
巨資四B, 04170217, 胡弘林

巨資四B, 04170228, 楊文瀚

巨量析2, 06571006, 蘇成恩

1 Introduction

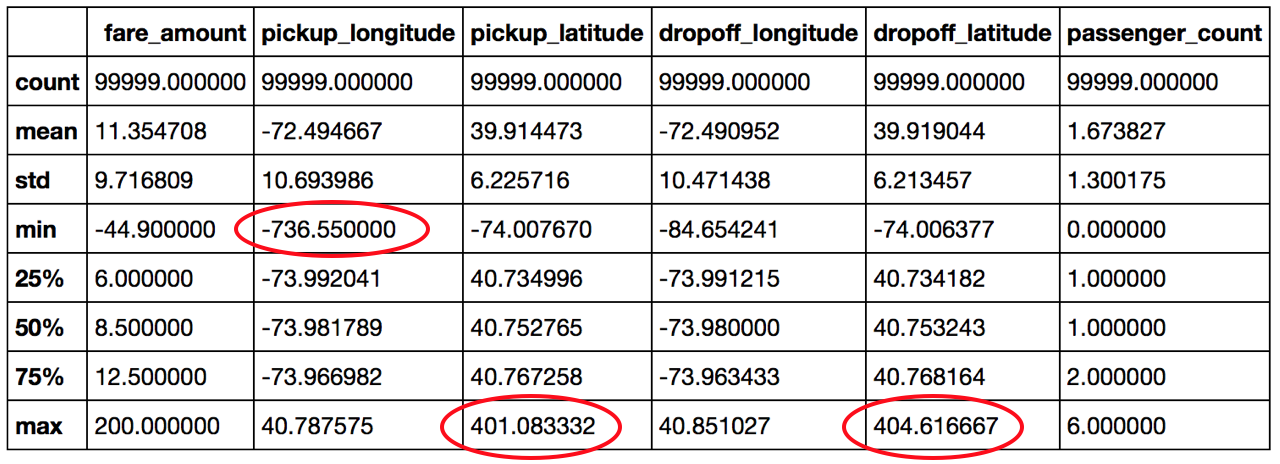
本報告書分析的資料為Kaggle上的New York City Taxi Fare Prediction競賽，Training data共有5千5百多萬筆資料，其中共有8個欄位，如下圖



分別為ID、車費、搭乘時間、搭乘地點經緯度、抵達地點經緯度以及載客數量，而最後要預測的變項為車費。為了節省分析時間以及電腦效能，因此我們切出Training data其中的10萬筆資料來做資料的前處理以及EDA，最後再套用回5000多萬筆的總資料。

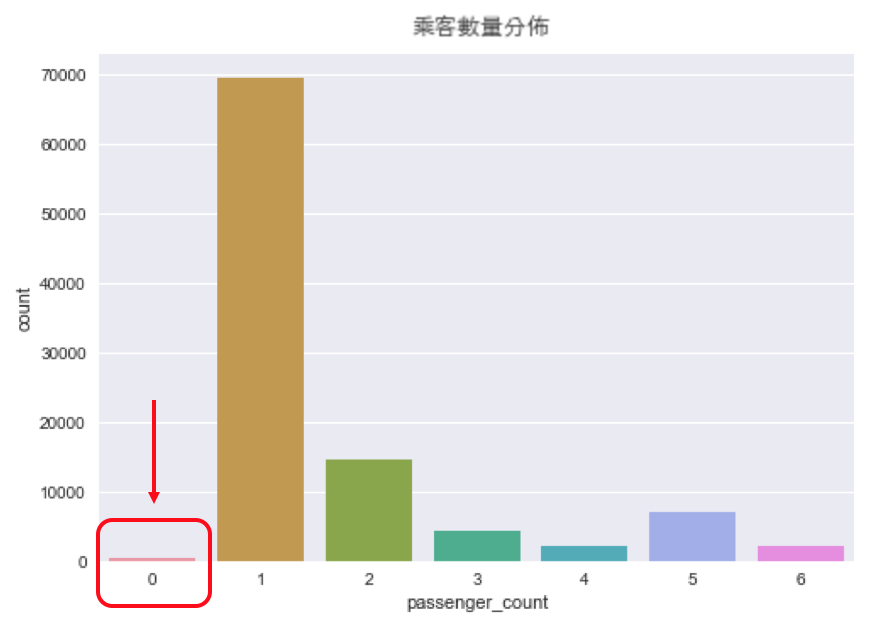
2 Exploratory data analysis for 10萬筆資料

1. 基本統計指標



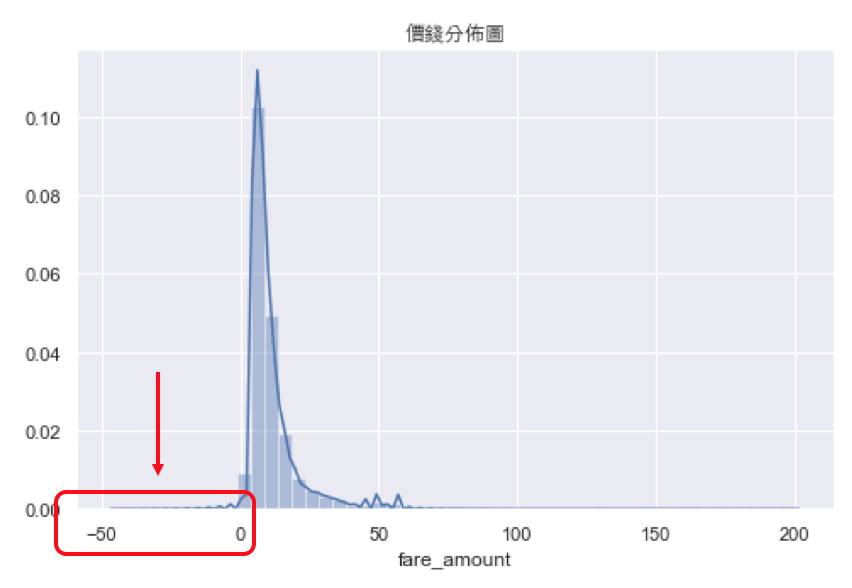
🡪 從上述可以發現在經緯度的資料當中存在異常值，經度(longitude)的範圍應在-180~180之間，而緯度(latitude)的範圍應在-90~90之間，因此我們首先去除資料中有異常經緯度的筆數，在10萬筆當中約有15筆異常值。

1. 載客數量分佈圖



🡪 乘客數量以1~6人為主要的分佈，不過也出現的0的少量異常值，初步猜想是人為輸入錯誤或是儀器錯誤。

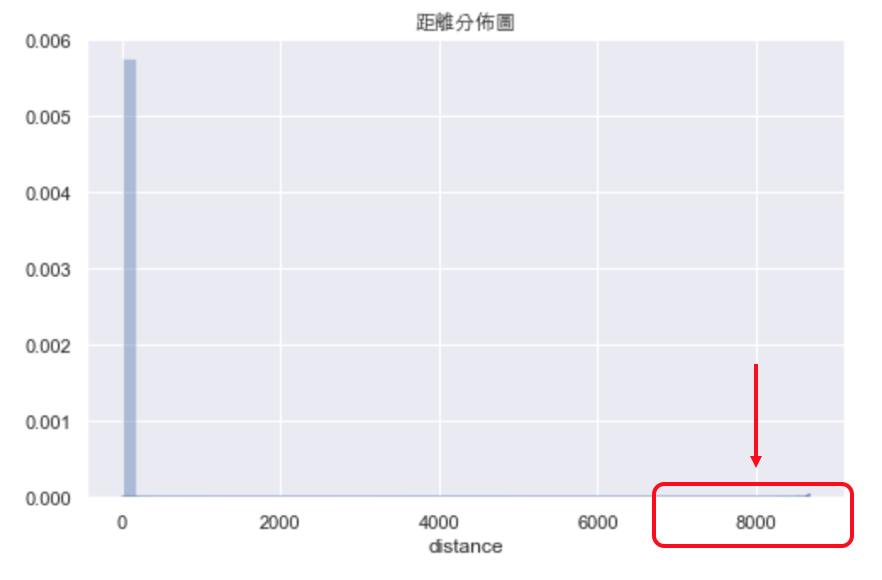
1. 價錢分佈圖



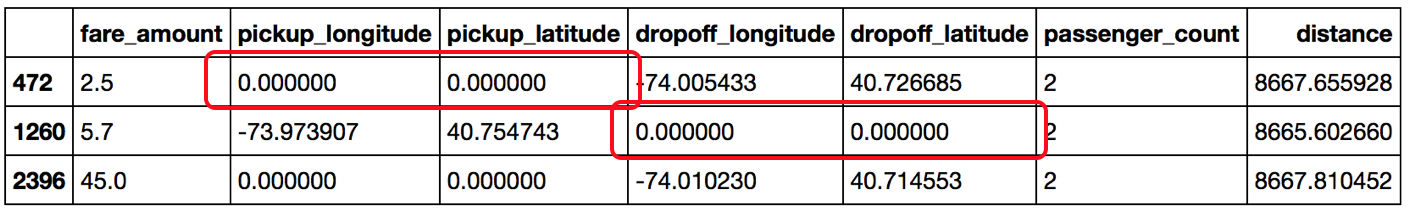
🡪 價錢分佈大多為0~20元左右，其中也出現的價格為負的異常值

1. 新增距離欄位(distance)

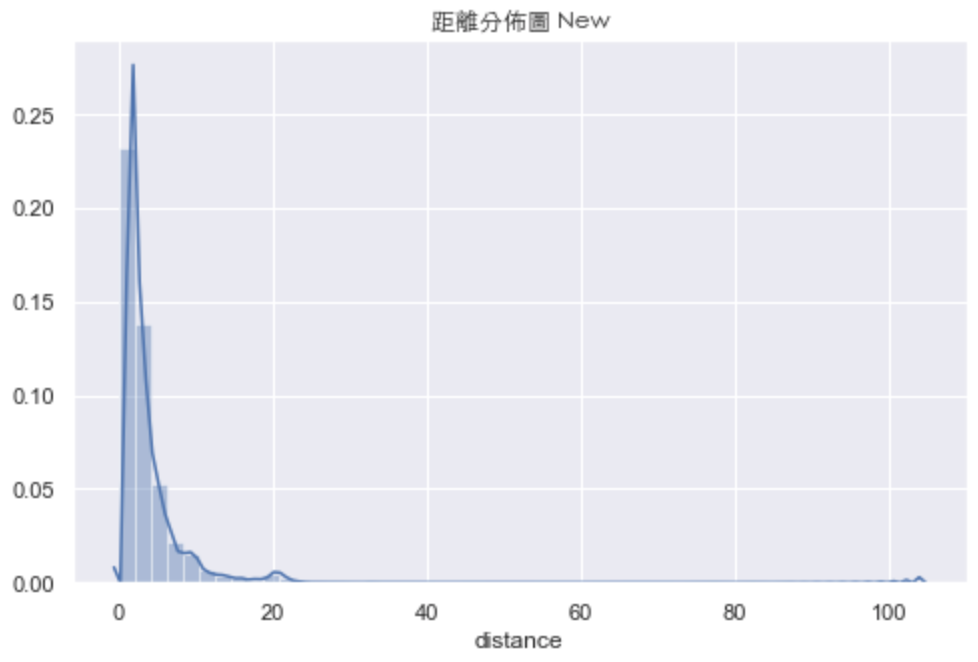
🡪 利用搭車地點的經緯度以及下車地點的經緯度做距離轉換，雖然算法是以兩者的直線距離為轉換結果，無法正確表達出計程車實際的行駛距離，但我們初步認為仍具有一定的參考性，就地點與地點間的直線距離越遠，價格越高，反之越近則價格越低。



🡪 查看距離的分佈圖後發現有距離異常高的狀況，反查資料後發現如下



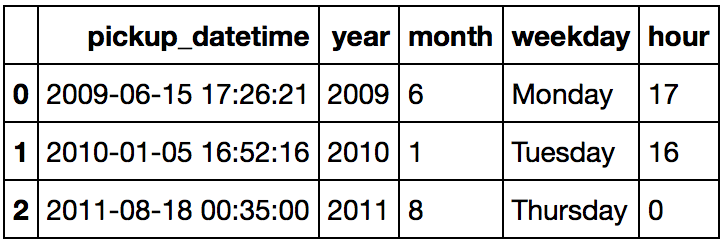
🡪 出現位於赤道的異常經緯度，因此對於經緯度的處理不能只有限制在正確的範圍內而已，我們的處理方式為先查看Testing data中的經緯度範圍為何，接著我們只要有包含到Testing data的經緯度資料就好，其他全部予以刪除，而我們限縮的經度範圍為-74.5~-72.8，而緯度範圍為40.5~41.8，如此一來不僅能去除定位錯誤的異常值，也能涵蓋Testing data的經緯度，下圖則為新的距離分佈圖



🡪 從上圖可以看出大部分搭乘的距離集中在5~15公里的範圍內

1. 新增時間欄位

🡪 我們也從搭車時間 (pickup\_datetime) 的欄位當中取出年份、月份、星期以及小時當作新的欄位變項，如下圖示例。



3  Data pre-processing for all data

1. 資料轉換

🡪 所有的Training data總共有5.7GB，匯入python當中總共需要1min 28s左右，不只緩慢也佔了不少的記憶體，因此我們將此份csv轉檔成feather格式，空間可以壓縮至1.6GB左右，減少記憶體的佔據，同時匯入python僅需23秒左右，增加分析的時間。

1. 刪除遺失值

🡪 原先的10萬筆資料中並無出現遺失值，而所有資料當中共出現了376筆遺失值，因此先予以刪除。

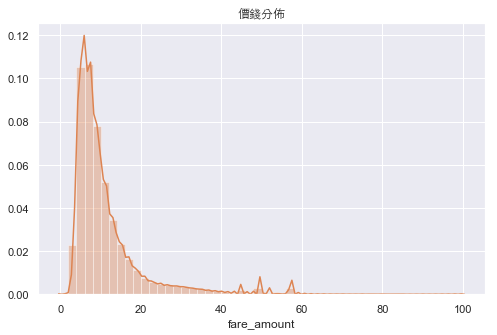
1. 限制乘客數量的範圍為1~6人

🡪 原先的10萬筆資料中有出現0人的情況，而在所有資料中又更加凌亂了，有出現200多人等的異常值，因此同樣予以刪除。

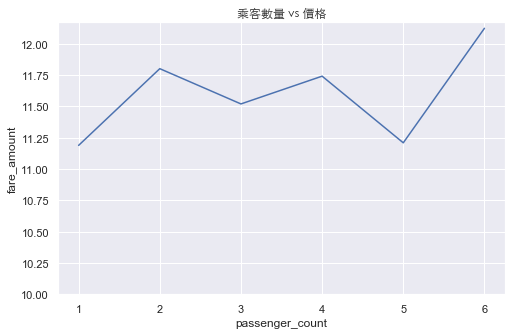
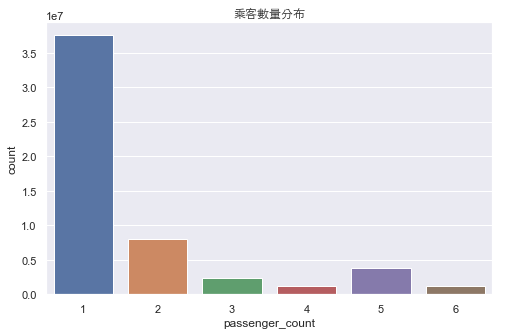
1. 刪除價格為負的資料
2. 限制經緯度範圍在 (-74.5, -72.8, 40.5, 41.8) 之間
3. 新增距離、年份、月份、星期、小時欄位變項

4  Exploratory data analysis for all data

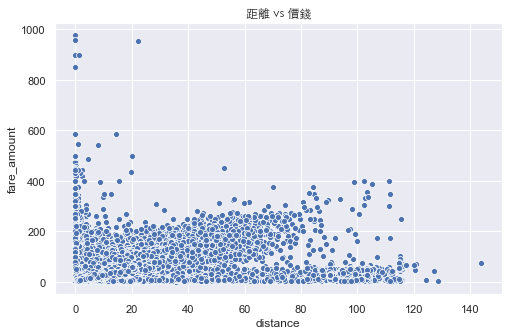
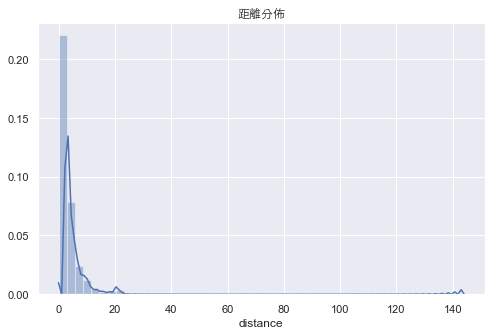
1. Feature 1 – 價格



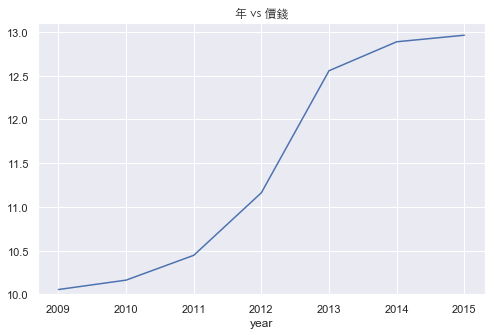
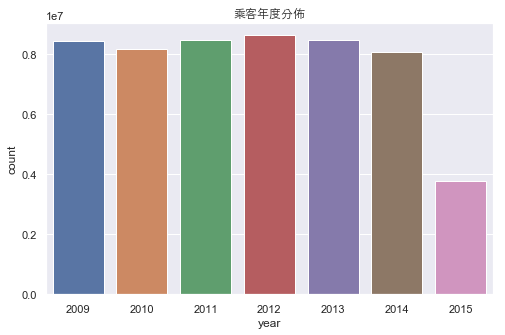
🡪 在所有資料中價錢分佈與10萬筆類似，大部分都是集中在0~20元之中

1. Feature 2 – 乘客數量

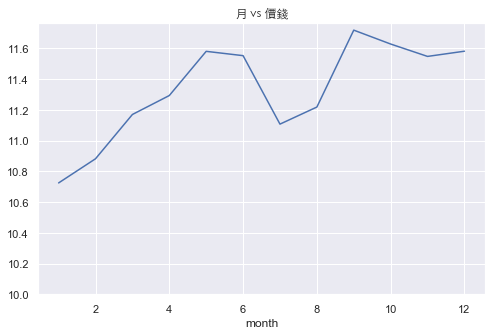
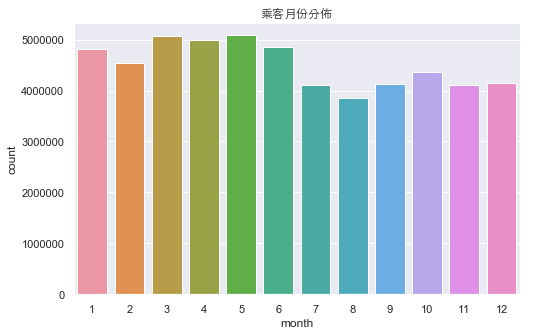
🡪 在左圖中我們可以看出此份資料大部分的乘客都是1人，而右圖可以看出載客數量的不同確實有影響到價格的不同

1. Feature 3 – 距離

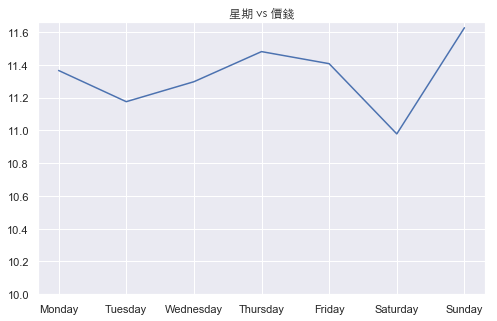
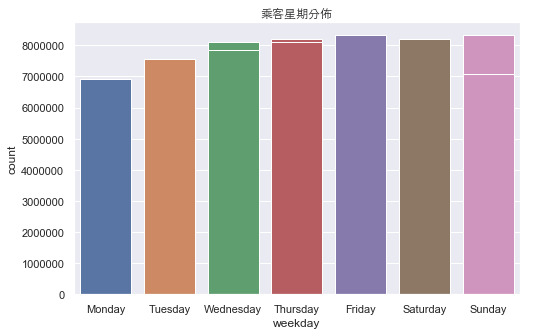
🡪 左圖與原先的10萬筆也類似，搭乘距離幾乎集中在20公里內左右，右圖則無法明顯看出距離與價格存在什麼樣的關係

1. Feature 4 – 年份

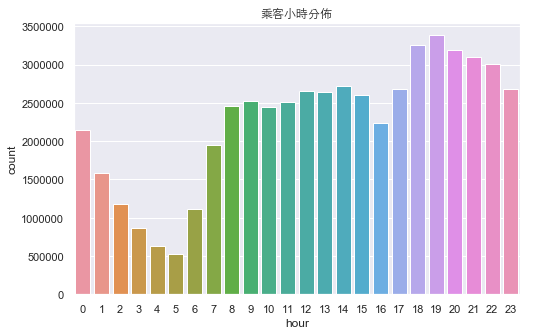
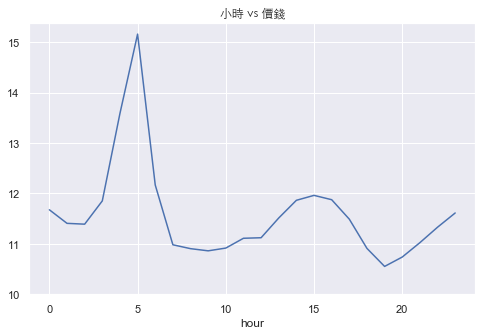
🡪 左圖可以看出個年度的乘車人數大致相同，無差距太大，2015年只剩一半是因為此份資料並無搜集完2015年的資料，而右圖明顯可以看出計程車的車資隨著年份而上漲，可以代表年份對於預測車資而言會是一個很好的變項。

1. Feature 5 – 月份

🡪 左圖可以看出搭乘計程車的人數在3~6月是比較多的，而右圖我們發現計程車車資也是會隨著月份的增加而大致上漲，唯獨在7月的時候平均價格較低，是一個有趣的觀點！同時也告訴我們月份對於價格也是存有影響力。

1. Feature 6 – 星期

🡪 左圖可以看出在星期一和二的人數相對較少，右圖則發現星期六時平均的車資為最低的時候，但星期日平均卻為最高。

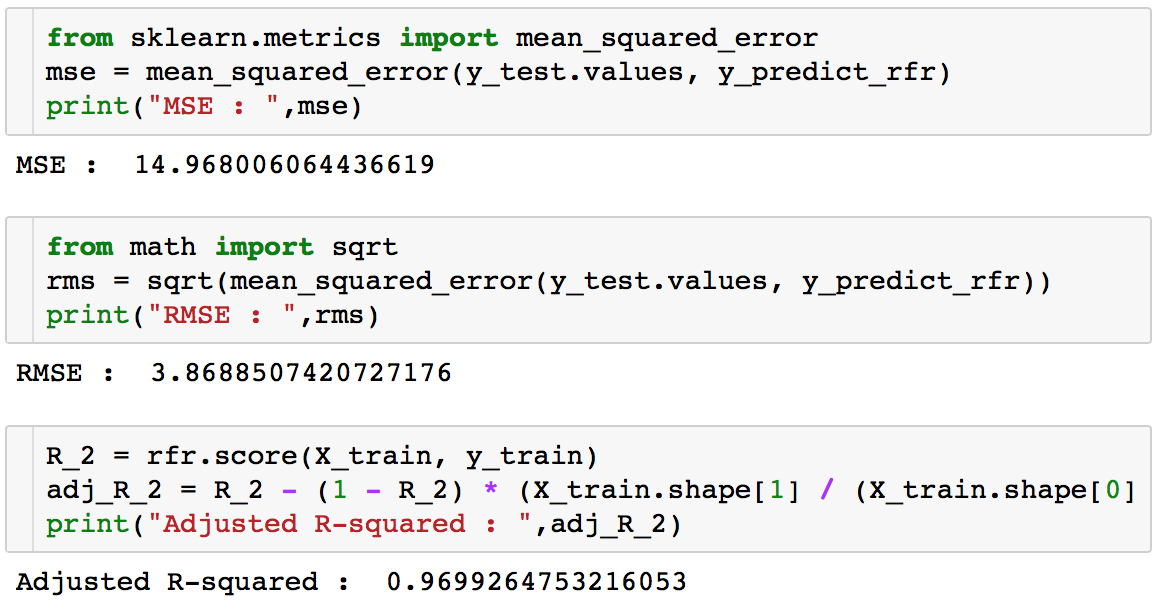
1. Feature 7 – 小時

🡪 左圖可以看出搭車人數在晚間6~8點時為最尖峰的時刻，凌晨則為最少，但從右圖中可以發現在凌晨5點之時的車資卻為最高，也是一個蠻有趣的觀點。

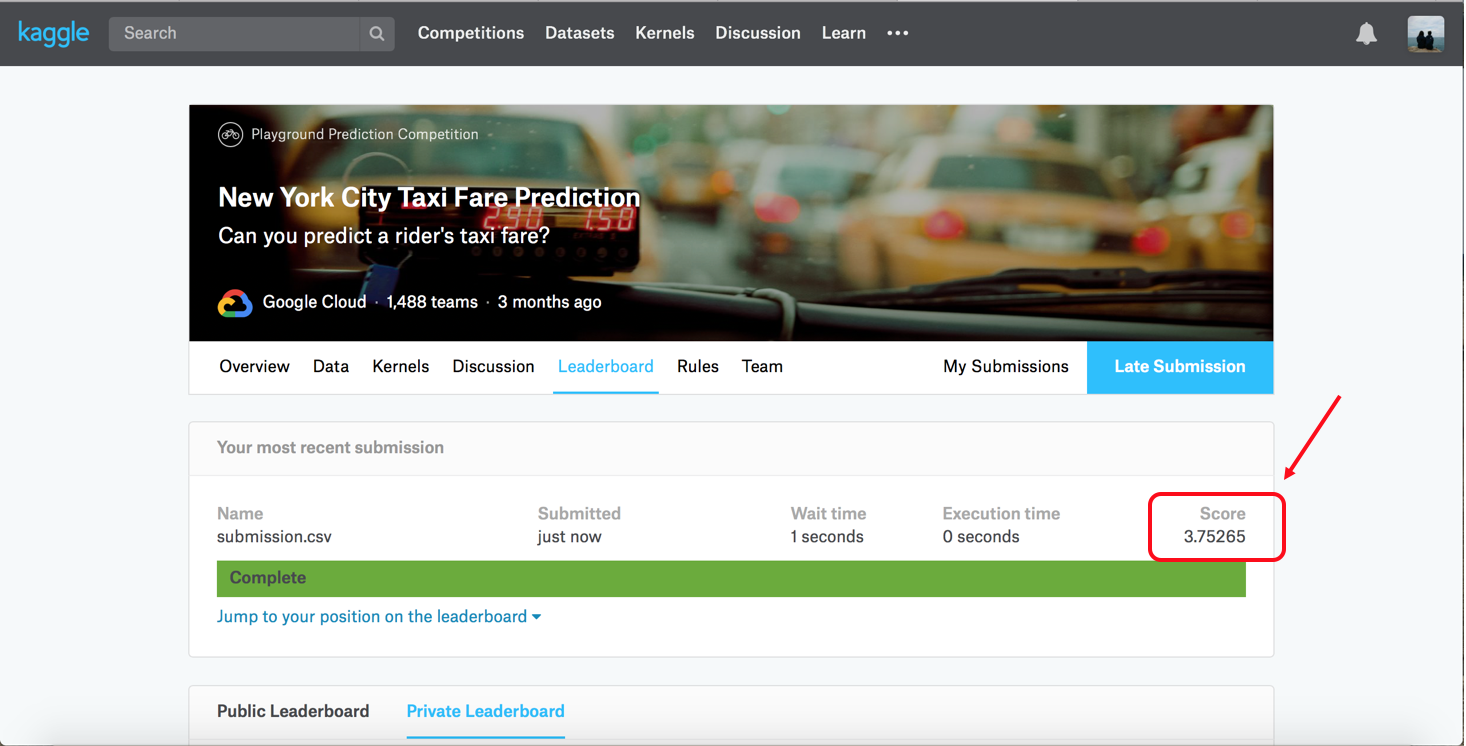
5  Data modeling

1. Model 1 – Random Forest Regression for 100萬training data

🡪 我們訓練過全部的5000多萬筆資料但發現電腦負載量過大，需花許多時間建成模型，因此先隨機挑選其中的100萬筆資料出來建置模型，至少可以先利用些指標來評估模型的好壞，模型建置大約需1min 30s左右，評估指標如下

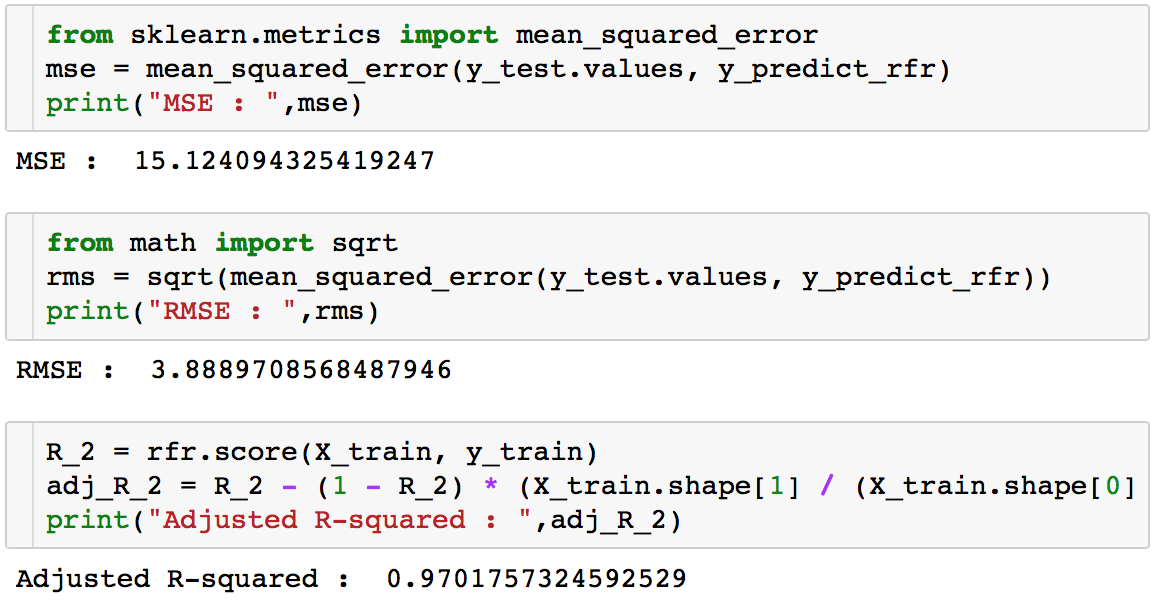


🡪 MSE和RMSE誤差因還沒有其他對照組比較，因此無法確定其好壞標準，但從調整後的判定係數來看，可以發現我們的變項篩選確實是很具有代表性的，透過這10個變項解釋力竟高達96%，我們也將第一版的模型預測testing data並丟上kaggle查看分數，而分數為3.75265，大約排名在800左右。

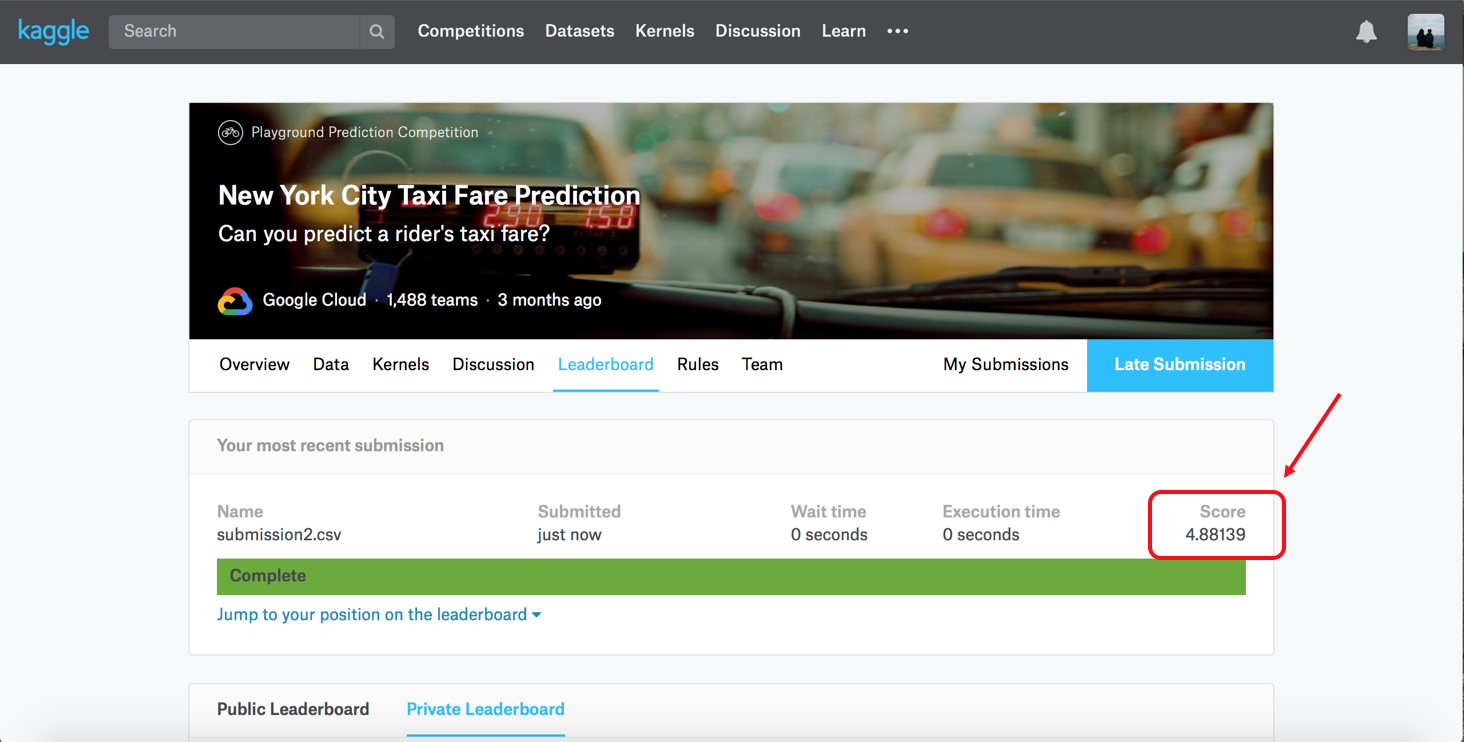


1. Model 2 – Random Forest Regression for 500萬training data

🡪 我們透過同樣的演算法新增訓練的資料至500萬筆，模型建置大約需9min 44s左右，模型評估指標如下



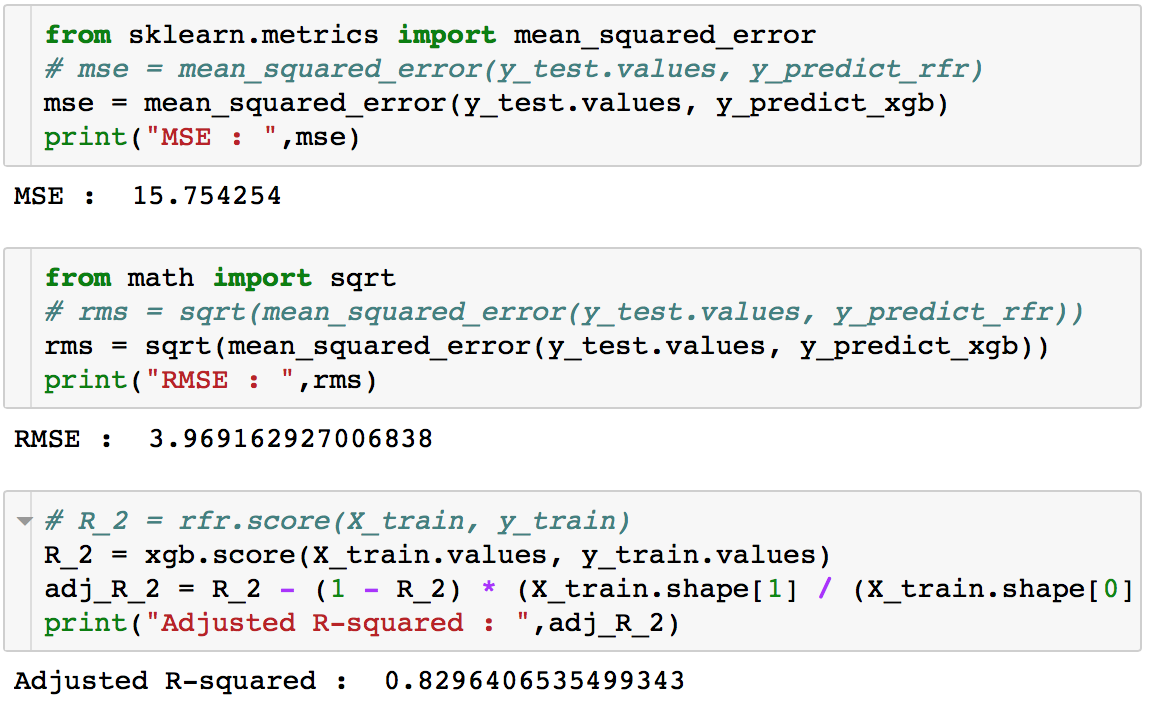
🡪 MSE和RMSE我們可以看到比原先的模型來的稍微高一些，與我們原先預期的會下降並不同，而在調整後的判定係數上仍維持96~97%的解釋力，與我們預期的相同，畢竟是使用同樣的變項建置模型，最後丟上kaggle查看分數



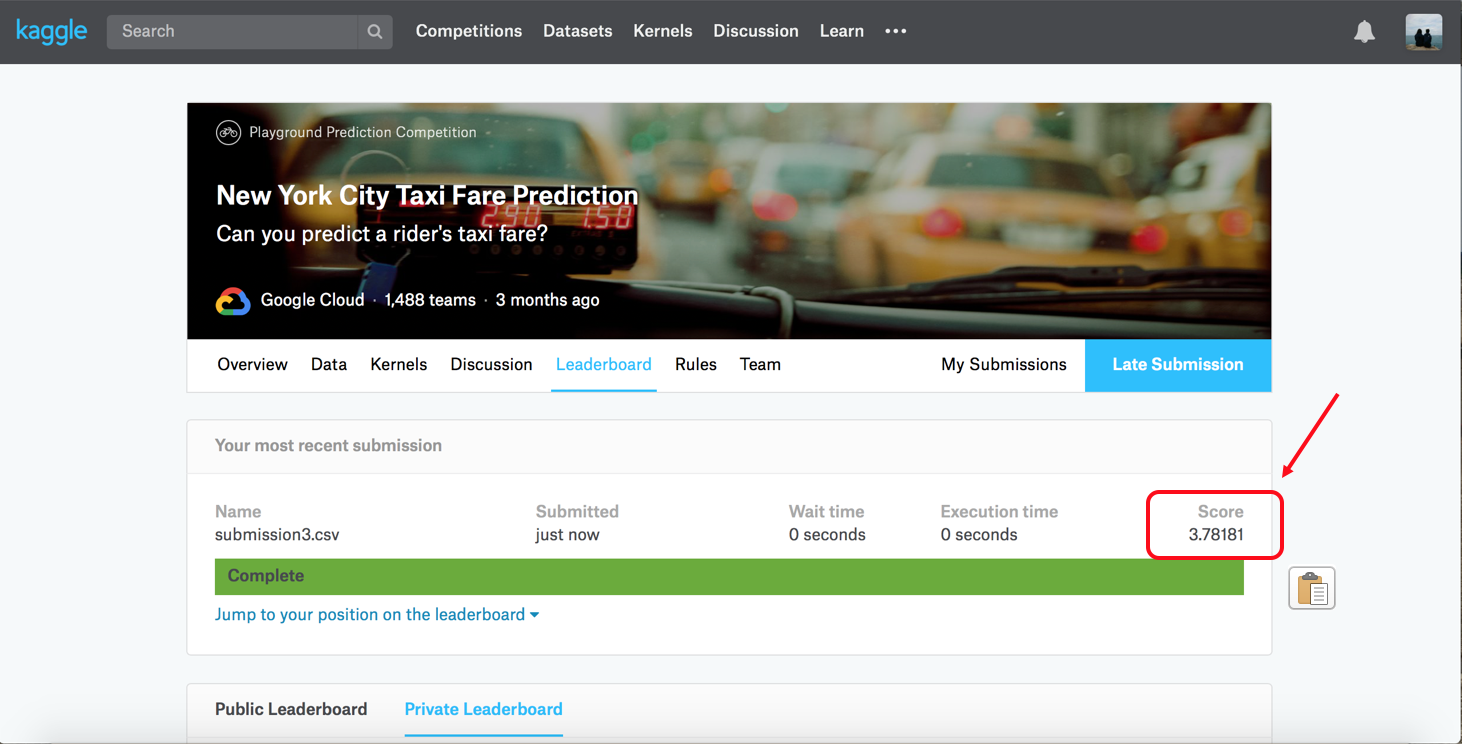
🡪 意外的是分數竟然比原先的還要低，而且低了不少，我們猜測是因為抽樣誤差的關係導致

1. Model 3– XGBoost Regression for 500萬training data

🡪 我們嘗試換了一種模型XGBoost Regression，訓練資料一樣給予500萬筆data，模型建置大約需8min 30s，模型評估指標如下

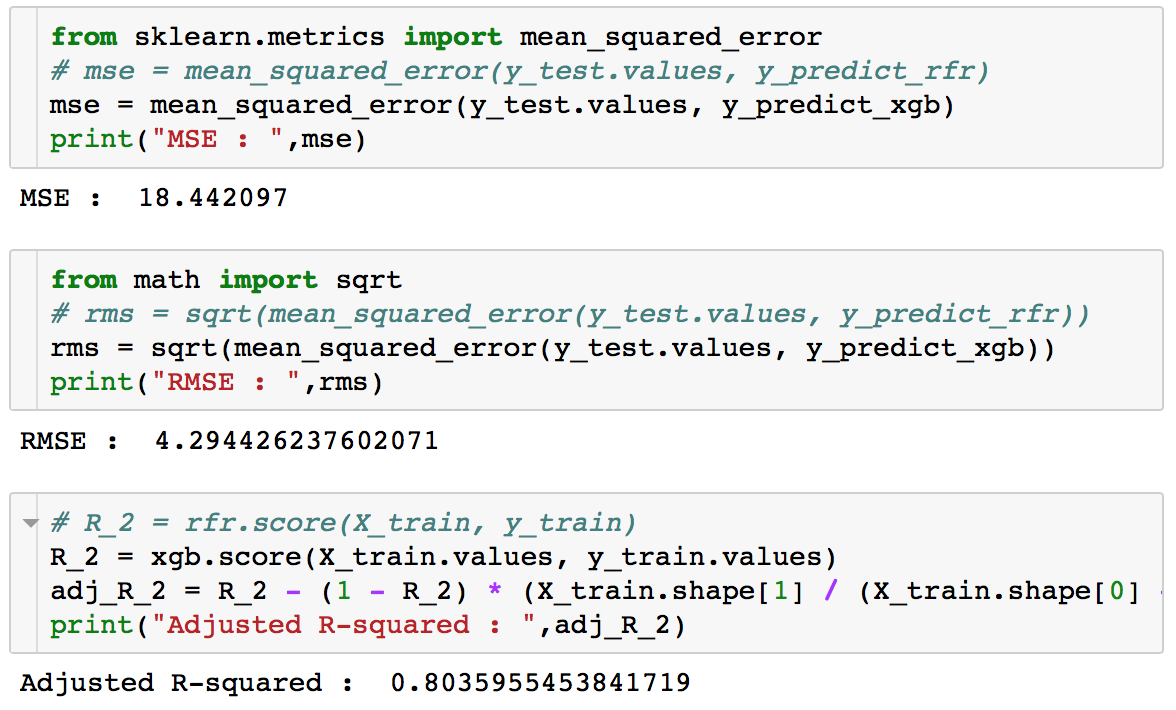


🡪 調整後的判定係數在XGBoost中表現的沒有比Random Forest Regression來的好，丟上kaggle後的分數為3.78181，也沒有模型1的效果來的好

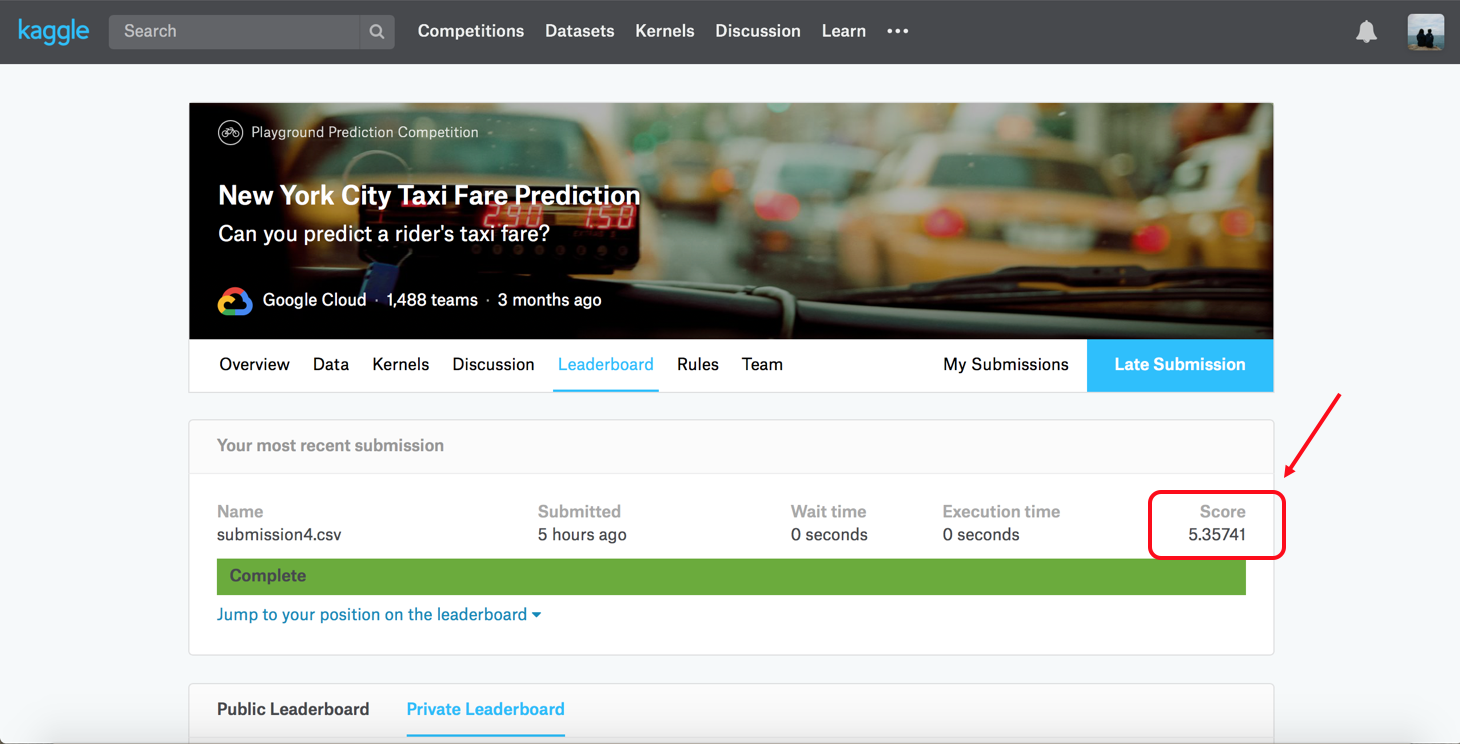


1. Model 4 – XGBoost Regression for所有5400萬data

🡪 我們在最後使用的所有的資料去訓練模型，總共花了約4個半小時完成，模型評估指標如下



🡪 我們發現調整後的判定係數在XGBoost中皆不如原本使用的Random Forest Regression，但Kaggle上的分數只有5.3左右令我們比較意外



6  Conclusion

🡪 我們在分析當中得出了幾個結論

1. 我們認為訓練比數的增加不代表能讓模型的預測能力越準確，訓練越多反而越有可能overfitting
2. 我們並未完全了解紐約市的計程車生態，也許像是機場到某個景點間是有固定的收費價格的，抑或是中間會經過固定的收費站等等，因此日後若要持續分析，我們並不能單單使用兩者的直線距離當變相而已，也許能匯入google地圖去判斷經緯度到經緯度之間會經過的路而算出真正距離，抑或是還有其他我們目前未發現到的車資規則也可以加入
3. 我們目前並未調整參數，也許能藉由調整模型參數而提高分數，而從我們的結果分析來看使用Random Forest Regression會是效果最好的演算法