1. 緒論

APP(Application)是指應用程式的意思，原本不限於個人電腦上或是手機上的應用程式，只是在這個人手一支智慧型手機年代，APP有時特指手機應用程式，APP的型態包羅萬象，包括社群軟體、遊戲、生產性工具、音樂、教育等等，族繁不及備載。APP的存在無疑豐富了人們的生活使人們的生活更加便利。

分析APP市場是一件重要的事情，根據Paolo Roma等人的研究，APP市場在2016年就超過了460億美元的營收，而在2020年1月8日的CNN報導中也指出Ios系統的消費者在短短一個禮拜內就消費了14億美元，種種數據顯示APP市場占據全球消費領域重要地位。

而APP的收費大致可以分成三種，一種是免費型的APP，一種是買斷型的APP，最後一種則是基本免費，但進階功能要收費的APP(以手機遊戲為主，也就是俗稱的儲值或是課金行為)，第一種的營收主要來自於廣告，但不管是哪一種APP，下載量均為APP開發商最關注的一環。

智慧型手機的作業系統主要分成兩種，第一種是以Apple公司開發的iOS作業系統，另外一種則是Google公司研發的Android作業系統，前者下載APP的商店稱為APP Store而後者下載APP的商店則稱為Google Play。根據Gartner公司的報告，上述兩種的作業系統在2016年合計占了智慧型手機99%的市佔率，所以可以知道分析上述兩種平台上的APP情形，便可以大致掌握APP市場。

本研究有兩個主要的資料集，一個是Google play商店的APP資料，另一個則是APP商店的APP資料，兩者均來自Kaggle(一個全球性的數據建模和數據分析競賽平台，並於2017年被Google公司收購)，

近年來有許多文獻分析APP的市場，包含分析APP製造商的投資策略、分析何種要素會影響APP的rating等等，但較少文獻在同時分析兩種平台的APP有何差別，是故本研究將會專注在這塊分析。

本研究將會專注在以下課題：

1. iOS的APP和Android的APP在APP的標題上用字偏好是否有不同的趨勢
2. 兩個平台影響rating的特徵是否有不同
3. 兩個平台影響定價的特徵是否有所不同
4. 分析Android APP市場的下載量和變數間的關係
5. 分析和預測Android市場的APP留言為正面還是負面

本研究的研究流程如下，首先會在第二章的文獻回顧中回顧有關APP市場相關的文獻，以及簡介使用的機器學習和計量模型方法，接著在第三章變數說明和敘述統計中，詳細說明兩個資料集的變數以及他們大致的資料分布，在第四章模型設計和實驗分析中，則會對相關問題使用適合的模型來分析，最後則在第五章的討論與建議中針對研究的不足給予建議。

隨機森林:

隨機森林為Leo Breiman於2001年提出的機器學習算法，它是一種集成模型，概念為透過平均或是多數決的方式來決定最後的答案，隨機森林的應用範圍很廣，可以應用在醫療、財金等多個領域，其優點為可以應用於高維度的資料，並且由於隨機抽取特徵的關係，每棵樹之間的相關性很低，所以可以避免過度配適的發生。

隨機森林的算法流程如下:

假設資料共有N個樣本以及M個特徵，首先要決定超參數(Hyper parameter)m和B，其中m為每次要抽取的特徵個數，B則為此森林總共要建立幾棵樹(tree)

首先透過bootstrap(重複放回抽樣)的方式抽出N個樣本作為每棵樹的訓練資料，接著隨機抽取m個特徵，之後從m個特徵中選取可以使當前節點的樣本的MSE降到最小的特徵作為樹的分支，並重複分支直到某些條件達成，例如下降的MSE小於某個特定數值，最後將B棵樹的結果去做平均或是投票得到最後的答案。

有別於一般衡量模型的好壞使用的是交叉驗證的方式，由於隨機森林採用的是樣本重複抽樣的方式採樣，是故在建構每棵樹的時候，都會有一些樣本沒有被抽到，這些樣本稱作OOB(out of bag sample)袋外樣本，原始資料中每個樣本從未被抽到的機率為(1-1/N)^N，當N趨近於無限大時，此極限機率約為36.8%，透過袋外樣本，即使我們的資料數不夠我們去切割測試集資料，我們也可以有效的衡量隨機森林模型表現的好壞。

袋外樣本的存在還有另一個好處，我們在線性回歸模型中可以透過T檢定等方式來檢測特徵是否重要，在隨機森林中，這件事依然可以輕鬆辦到。

首先先記錄每個樣本的OOB誤差，接著將其在整個森林上做平均，假使我們想要檢驗第j個特徵是否是重要變數的話，我們可以隨機打亂第j個特徵，並重新計算打亂後的資料的OOB誤差。第j個特徵的重要性則可以透過打亂前後的OOB誤差差值平均來得到，最後透過將插值的平均做標準化，便可以得到此變數的重要性分數。此做法背後的想法也相當直覺，如果該變數真的如此重要的話，那麼隨便亂打亂的話，打亂前後的OOB誤差便要相差很大，如果相差不大，那就代表此變數不重要。

交叉驗證:

常見的機器學習方法都會有一個超參數需要在模型使用前被決定，例如在最近鄰居法當中，便需要決定要使用幾個鄰居來當作投票或是平均的依據，鄰居數K，便是最近鄰居法的超參數。

交叉驗證的主要用途便是可以幫忙決定模型的超參數，以最常見的5 fold cross validation來舉例，將原始資料盡量平均的分割成五份，其中每次選取一份資料作為測試集測試模型，用其餘四份資料作為訓練集來訓練模型，所以上述的動作總共會做五次，每一分資料都會恰有一次作為測試集來驗證資料，將上述五次的誤差做平均，接著選取會使平均誤差最小的超參數值當成最好的超參數值。

Lasso:

Robert Tibshirani於1996年提出了Lasso(least absolute shrinkage and selection operator)的算法，該模型最早應用於線性回歸，該模型基本態樣如下:

顯而易見的，在lambda為零的狀況下，lasso regression退化為一般的線性模型，在lambda逐漸增大時，會有越來越多的變數係數值會直接變成0。

由於標準的線性回歸無法處理特徵比樣本數多的時候的狀況，Lasso的算法透過施予L1 norm的懲罰項，會強迫一些不重要的變數的係數值變成0，因此使得線性估計可行，更重要的是我們也因此找到的重要的變數。

Lasso不只可以應用在線性回歸模型，也可以應用在廣義線性模型，例如邏輯斯回歸等

詞袋模型:

詞袋模型是一個在自然語言處理當中一個相當簡單的模型，其概念為建立一個X matrix 維度為m\*n，代表著有m則不同的文章或是留言和n個不同的詞彙，其元素Xij代表著第j個詞彙在第i個文章當中出現的次數。

計量相關模型:

Tobit model:

Ordinal logistic regression:

APP相關的文獻:

Ahsan(2019)指出使用者通常喜歡下載評價高的APP，因為評價高的APP反映了使用者的滿意度，而APP開發商為了使APP獲得更高的評價，會使用各式各樣不同的技巧，例如更吸引人的APP標題或是更吸引人的APP icon等方式，但是並沒有一個明確的方式可以找出何種因子真正的影響APP的評價，是故它使用了一筆樣本數10840的Google play 資料集透過線性回歸、隨機森林和支持向量回歸這三種機器學習的方式找出影響APP評價的重要變數，他的研究指出:

評論的留言數、APP的容量大小、以及APP的下載數為最影響APP的變數，他也另外做了針對APP的標題的文字分析，根據他的分析，標題中含有workout、notes和space的APP評價最高，其平均評價分別為4.61、4.5和4.47。

Gunwoog(2011)等人則指出了APP store APP開發商的投資策略，他們關注的課題為開發商能有多少個APP在下載量前300，他們透過配適poission回歸中發現，相對於專一研發在特定項目的APP開發商比起來，研發多種項目的APP開發商是更好的投資策略，以及與其只開發付費APP，同時開發免費和付費APP的開發商有更好的營收表現。

Bin FU(2013)等人則專注在APP的評論留言和APP的評價之間的關係，他們收集了171493個不同的APP以及超過1300萬則留言，經過刪除了html 標籤和刪除了非英語的留言後，他們構造了一個維度為988960\*19387的詞袋模型矩陣，並將此丟入帶有Elastic Net懲罰項的回歸模型，他們最大的貢獻在於由於有帶有懲罰項的回歸，所以他們藉由找出重要的詞彙後，進一步去偵測出留言TRI(Text-Rating-Inconsistency)現象，亦即留言文字內容和評論並不一致的狀況。