(一) 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。

APP 相關的文獻:

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