

國立清華大學
智慧化企業整合
期中報告

Kaggle 房價預測競賽

指導教授：邱銘傳博士

組別：8

學生：楊怡芳 109034508 陳譽升 109034538

鄭建澤 109034539 胡心玫 109034540

一、 背景介紹

1. 情境描述

今年巧遇肺炎疫情挑戰，各國央行為因應此衝擊，都在積極地進行量化寬鬆政策，因此市場流入大量熱錢，導致房地產與股市皆可樂觀看待。如今房價狂漲，我們需要一種方式針對特定房地產物件的價值進行判斷，以期能夠正確的估算該物件的真實價值，而非隨市場泡沫隨意喊價，成為房地產的冤大頭。因此，透過上課所學之深度學習技法，希望能夠透過房地產物件的環境條件等等因素，成功預測出該物件的真實價值，造福民眾。

2. 問題定義

我們上網搜尋房價預測資料集進行探討，發現 Kaggle 從 2016 年 8 月開始，到 2017 年 2 月舉辦關於房價預測競賽，決定以此作為本次報告之主題。而此次比賽提供 79 個特徵欄位來描述美國愛荷華州住宅的各式資訊，參賽者須從中挑出重要的特徵訓練模型並預測該房屋的售價，以下將透過 5W1H 手法進行問題定義分析：

What?	美國愛荷華州住宅房價預測
When?	欲購買、販賣住宅與了解房地產行情時
Who?	想擁有自家住宅或欲販售住宅之民眾
Where?	美國愛荷華州
Why?	市場泡沫，房地產價值狂漲，導致人民難以預測房價行情
How?	資料欲處理/視覺化、機器學習、深度學習

表 1 - 房價預測 5W1H 問題定義

二、 資料集描述

資料集分為兩個檔案：train.csv、test.csv，前者檔案包含 1460 筆資料，而後者則是競賽的測試資料集，共 1459 筆資料。兩者皆包含 79 個房屋特徵資訊，但 train.csv 多了 SalePrice 這個欄位，以做為訓練模型的答案驗證，參賽者須利用 train.csv 訓練好的模型，來預測 test.csv 的 SalePrice 結果，並將該結果上傳 kaggle 後，該平台即會提供一個分數來衡量模型好壞。

feature	description	type
SalePrice	the property's sale price in dollars (target variable that you're trying to predict)	continuous
LotFrontage	Linear feet of street connected to property	
LotArea	Lot size in square feet	
MasVnrArea	Masonry veneer area in square feet	
BsmtFinSF1	Type 1 finished square feet	
BsmtFinSF2	Type 2 finished square feet	

BsmtUnfSF	Unfinished square feet of basement area	continuous
TotalBsmtSF	Total square feet of basement area	
1stFlrSF	First Floor square feet	
2ndFlrSF	Second floor square feet	
LowQualFinSF	Low quality finished square feet (all floors)	
GrLivArea	Above grade (ground) living area square feet	
BsmtFullBath	Basement full bathrooms	
BsmtHalfBath	Basement half bathrooms	
FullBath	Full bathrooms above grade	
HalfBath	Half baths above grade	
BedroomAbvGr	Number of bedrooms above basement level	
KitchenAbvGr	Number of kitchens	
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)	
Fireplaces	Number of fireplaces	
GarageCars	Size of garage in car capacity	
GarageArea	Size of garage in square feet	
WoodDeckSF	Wood deck area in square feet	
OpenPorchSF	Open porch area in square feet	
EnclosedPorch	Enclosed porch area in square feet	
3SsnPorch	Three season porch area in square feet	
ScreenPorch	Screen porch area in square feet	
PoolArea	Pool area in square feet	
MiscVal	\$Value of miscellaneous feature	
YearBuilt	Original construction date	
YearRemodAdd	Remodel date	
GarageYrBlt	Year garage was built	
MSSubClass	The building class	
MSZoning	The general zoning classification	
Street	Type of road access	
Alley	Type of alley access	
LotShape	General shape of property	
LandContour	Flatness of the property	
Utilities	Type of utilities available	
LotConfig	Lot configuration	
LandSlope	Slope of property	
Neighborhood	Physical locations within Ames city limits	
Condition1	Proximity to main road or railroad	
Condition2	Proximity to main road or railroad (if a second is present)	
BldgType	Type of dwelling	
HouseStyle	Style of dwelling	
OverallQual	Overall material and finish quality	

OverallCond	Overall condition rating	category
RoofStyle	Type of roof	
RoofMatl	Roof material	
Exterior1st	Exterior covering on house	
Exterior2nd	Exterior covering on house (if more than one material)	
MasVnrType	Masonry veneer type	
ExterQual	Exterior material quality	
ExterCond	Present condition of the material on the exterior	
Foundation	Type of foundation	
BsmtQual	Height of the basement	
BsmtCond	General condition of the basement	
BsmtExposure	Walkout or garden level basement walls	
BsmtFinType1	Quality of basement finished area	
BsmtFinType2	Quality of second finished area (if present)	
Heating	Type of heating	
HeatingQC	Heating quality and condition	
CentralAir	Central air conditioning	
Electrical	Electrical system	
KitchenQual	Kitchen quality	
Functional	Home functionality rating	
FireplaceQu	Fireplace quality	
GarageType	Garage location	
GarageFinish	Interior finish of the garage	
GarageQual	Garage quality	
GarageCond	Garage condition	
PavedDrive	Paved driveway	
PoolQC	Pool quality	
Fence	Fence quality	
MiscFeature	Miscellaneous feature not covered in other categories	
MoSold	Month Sold	
YrSold	Year Sold	
SaleType	Type of sale	
SaleCondition	Condition of sale	

表 2 - 欄位特徵敘述表

三、 特徵挑選/資料前處理

資料集提供的特徵數高達 79 個，但不是每個變數都與 Saleprice 有關，所以我們做了以下觀察分析，最終僅挑選出 54 個變數做為 training data。

1. 觀察缺失值

統計 train 跟 test data 的缺失值後發現，有 5 個變數的缺失值超過資料的半數，故不予納入訓練，下表臚列五項欄位的缺失資料筆數。

	Train (共 1460 筆)	Test (共 1459 筆)
Alley	1369	1352
FireplaceQu	690	730
PoolQC	1453	1456
Fence	1179	1169
MiscFeature	1406	1408

表 3 – 存在過多缺失值之特徵缺失資料數欄位表

2. 相關性比較



圖 1 – 不同資料型態之自變數與依變數統計分析方法

在進行模型訓練前需先瞭解訓練特徵是屬於何種資料型態，以利計算各特徵欄位與目標變數的關係。根據表 2 可知，預測目標 SalePrice 為連續型變數，而經第一步剔除過多缺失值之欄位後，有將近六成為類別型變數(continuous:categorical = 32:43)，故我們將參照圖 1 利用「相關分析、T 檢定」進行各特徵欄位與目標變數的相關性研究，以利模型訓練之特徵挑選。

I. 目標變數與自變數皆為「連續型」

利用 python pandas 提供的套件 `pandas.DataFrame.corr(method = pearson)` 來計算兩兩的相關性。得出 SalePrice 與其他連續型變數之相關係數如表 3，由表可知「GrLivArea」與 SalePrice 具高度正相關，可當模型訓練的特徵之一；而雖然「GarageArea、TotalBsmtSF、1stFlrSF」未達高度相關之標準，但全部皆有 0.6 的表現，故我們將針對這 3 個變數做更深入的視覺化分析，以決定後續要將那些變數納入訓練考量。

	SalePrice		
GrLivArea	0.708624	OpenPorchSF	0.315856
GarageArea	0.623431	HalfBath	0.284108
TotalBsmtSF	0.613581	LotArea	0.263843
1stFlrSF	0.605852	BsmtFullBath	0.227122
FullBath	0.560664	BsmtUnfSF	0.214479
TotRmsAbvGrd	0.533723	BedroomAbvGr	0.168213
YearBuilt	0.522897	ScreenPorch	0.111447
YearRemodAdd	0.507101	PoolArea	0.092404
GarageYrBlt	0.486362	3SsnPorch	0.044584
MasVnrArea	0.477493	BsmtFinSF2	-0.01138
Fireplaces	0.466929	BsmtHalfBath	-0.01684
BsmtFinSF1	0.38642	MiscVal	-0.02119
LotFrontage	0.351799	LowQualFinSF	-0.02561
WoodDeckSF	0.324413	EnclosedPorch	-0.12858
2ndFlrSF	0.319334	KitchenAbvGr	-0.13591

表 3 - SalePrice 與其他連續型變數相關係數表

II. 目標變數為「連續型」，自變數為「類別型」

利用 R 將所有變數與 SalePrice 建立一個迴歸模型，得出的 r-squared 為 0.86、adjusted r-squared 為 0.84，顯示該模型具有一定的解釋力，而模型算出各類別變數與 SalePrice 之 t 分數如表 4，由表可知，t 分數的絕對值越大，該變數對模型影響越顯著(越 significant * 越多，***代表 P-value < 0.001，**代表 P-value < 0.01，*代表 P-value < 0.1)，跟 SalePrice 可能越有關係，而因 kaggle 提供的筆數僅有 1460 筆，故我們將超過 2 顆*的變數皆納入我們的訓練因子。

	t value	significant			
MSZoningRL	2.139	*	NeighborhoodIDOTRR	-2.235	*
LotShapeIR3	2.387	*	NeighborhoodMitchel	-2.55	*
LotConfigCulDSac	2.185	*	NeighborhoodNAMES	-2.573	*
LotConfigInside	-2.23	*	NeighborhoodNridgHt	2.08	*
LandSlopeMod	2.027	*	NeighborhoodOldTown	-2.375	*
NeighborhoodCollgCr	-2.07	*	NeighborhoodSawyer	-2.067	*
NeighborhoodGilbert	-2.323	*	Condition1Norm	2.012	*
			Condition1PosN	2.171	*

Condition2PosA	2.201	*
HouseStyle1.5Unf	-2.226	*
HeatingQCGd	-2.393	*
FunctionalMaj2	-2.237	*
GarageFinishUnf	-2.037	*
GarageQualGd	-2.534	*
GarageQualPo	-2.564	*
GarageCondGd	2.51	*
SaleConditionAlloca	2.369	*
SaleConditionNormal	2.074	*
LotConfigFR2	-3.067	**
NeighborhoodEdwards	-2.815	**
BsmtExposureNo	-2.676	**
HeatingQCTA	-2.629	**
GarageTypeNoGRG	-3.056	**
GarageQualFa	-2.973	**
GarageQualTA	-2.806	**
GarageCondFa	2.632	**
GarageCondPo	2.729	**
GarageCondTA	2.736	**
NeighborhoodNoRidge	5.955	***
NeighborhoodStoneBr	3.7	***
Condition2PosN	-4.103	***
BldgTypeTwnhs	-3.441	***
BldgTypeTwnhsE	-3.58	***
HouseStyle1Story	-3.393	***
HouseStyleSFoyer	-5.052	***
HouseStyleSLvl	-3.57	***
OverallQual	12.204	***
RoofMatlCompShg	7.736	***
RoofMatlMembran	4.28	***
RoofMatlMetal	4.203	***
RoofMatlRoll	5.284	***
RoofMatlTar&Grv	6.218	***
RoofMatlWdShake	7.029	***
RoofMatlWdShngl	9.458	***
ExterQualGd	-3.776	***
ExterQualTA	-3.666	***
BsmtQualFa	-4.865	***
BsmtQualGd	-7.453	***
BsmtQualTA	-7.205	***

BsmtExposureGd	6.204	***
BsmtFinType1Unf	-3.497	***
KitchenQualFa	-4.718	***
KitchenQualGd	-6.777	***
KitchenQualTA	-7.385	***
(Intercept)	1.704	.
MSZoningFV	1.815	.
MSZoningRH	1.791	.
MSZoningRM	1.905	.
LandContourLvl	1.672	.
HouseStyle2.5Fin	1.705	.
OverallCond	1.74	.
Exterior1stlmStucc	-1.885	.
FoundationPConc	1.668	.
BsmtFinType1LwQ	-1.708	.
YrSold	-1.81	.
SaleTypeNew	1.759	.
MSSubClass	-0.26	.
StreetPave	0.891	.
LotShapeIR2	1.32	.
LotShapeReg	-0.579	.
LandContourHLS	0.362	.
LandContourLow	-0.24	.
UtilitiesNoSeWa	-1.253	.
LotConfigFR3	-1.029	.
LandSlopeSev	1.22	.
NeighborhoodBlueste	-0.356	.
NeighborhoodBrDale	-1.11	.
NeighborhoodBrkSide	-1.508	.
NeighborhoodClearCr	-0.632	.
NeighborhoodCrawfor	0.911	.
NeighborhoodMeadowV	-0.803	.
NeighborhoodNPkVill	0.139	.
NeighborhoodNWAmes	-1.316	.
NeighborhoodSawyerW	-0.848	.
NeighborhoodSomerst	-0.329	.
NeighborhoodSWISU	-1.201	.
NeighborhoodTimber	-1.159	.
NeighborhoodVeenker	0.337	.
Condition1Feedr	0.716	.
Condition1PosA	1.459	.

Condition1RR Ae	-0.411	
Condition1RR An	0.086	
Condition1RR Ne	-0.71	
Condition1RR Nn	-0.795	
Condition2Feedr	0.074	
Condition2Norm	-0.112	
Condition2RR Ae	0.012	
Condition2RR An	-0.015	
Condition2RR Nn	0.197	
BldgType2fmCon	0.105	
BldgTypeDuplex	1.131	
HouseStyle2.5Unf	0.039	
HouseStyle2Story	0.181	
RoofStyleGable	-0.364	
RoofStyleGambrel	0.232	
RoofStyleHip	-0.039	
RoofStyleMansard	0.331	
RoofStyleShed	-0.172	
Exterior1stAsphShn	0.201	
Exterior1stBrkComm	-0.802	
Exterior1stBrkFace	1.508	
Exterior1stCBlock	-1.093	
Exterior1stCemntBd	1.599	
Exterior1stHdBoard	-0.718	
Exterior1stMetalSd	0.226	
Exterior1stPlywood	0.006	
Exterior1stStone	0.546	
Exterior1stStucco	0.614	
Exterior1stVinylSd	-0.725	
Exterior1stWd Sdng	-0.246	
Exterior1stWdShng	-0.662	
Exterior2ndAsphShn	-0.554	
Exterior2ndBrk Cmn	0.141	
Exterior2ndBrkFace	-0.622	
Exterior2ndCBlock	NA	
Exterior2ndCmentBd	-1.446	
Exterior2ndHdBoard	0.152	
Exterior2ndImStucc	1.373	
Exterior2ndMetalSd	-0.553	
Exterior2ndOther	-0.203	
Exterior2ndPlywood	0.002	

Exterior2ndStone	-0.804	
Exterior2ndStucco	-0.434	
Exterior2ndVinylSd	0.352	
Exterior2ndWd Sdng	-0.268	
Exterior2ndWd Shng	-0.767	
MasVnrTypeBrkFace	0.215	
MasVnrTypeNone	-0.475	
MasVnrTypeStone	0.748	
ExterQualFa	-1.015	
ExterCondFa	-0.08	
ExterCondGd	0.593	
ExterCondPo	-0.098	
ExterCondTA	0.476	
FoundationCBlock	1.477	
FoundationSlab	0.109	
FoundationStone	1.521	
FoundationWood	-0.024	
BsmtQualNoBSMT	-1.514	
BsmtCondGd	-1.035	
BsmtCondNoBSMT	NA	
BsmtCondPo	-0.006	
BsmtCondTA	0.028	
BsmtExposureMn	-0.507	
BsmtExposureNoBSMT	-0.562	
BsmtFinType1BLQ	-0.639	
BsmtFinType1GLQ	1.453	
BsmtFinType1NoBSMT	NA	
BsmtFinType1Rec	-0.57	
BsmtFinType2BLQ	-0.878	
BsmtFinType2GLQ	-0.077	
BsmtFinType2LwQ	-0.558	
BsmtFinType2NoBSMT	0.667	
BsmtFinType2Rec	-0.393	
BsmtFinType2Unf	-1.054	
HeatingGasA	0.242	
HeatingGasW	0.877	
HeatingGrav	0.161	
HeatingOthW	0.053	
HeatingWall	0.338	
HeatingQCFa	-1.516	
HeatingQCPo	0.008	

CentralAirY	1.526	
ElectricalFuseF	-0.349	
ElectricalFuseP	0.042	
ElectricalMix	0.788	
ElectricalSBrkr	0.083	
FunctionalMin1	-0.567	
FunctionalMin2	-0.422	
FunctionalMod	0.422	
FunctionalSev	-1.139	
FunctionalTyp	-0.839	
GarageTypeAttchd	-0.074	
GarageCars	-0.068	
GarageTypeBasment	-0.227	
GarageTypeBuiltIn	0.548	
GarageTypeCarPort	-0.095	
GarageTypeDetchd	-0.252	
GarageFinishNoGRG	NA	

GarageFinishRFn	-1.559	
GarageQualNoGRG	NA	
PavedDriveP	-0.062	
PavedDriveY	0.25	
MoSold	-1.499	
SaleTypeCon	0.771	
SaleTypeConLD	0.95	
SaleTypeConLI	-0.064	
SaleTypeConLw	-1.025	
SaleTypeCWD	-0.11	
SaleTypeOth	0.04	
SaleTypeWD	-1.521	
SaleConditionAdjLand	-0.234	
SaleConditionFamily	-0.184	
SaleConditionPartial	-1.008	

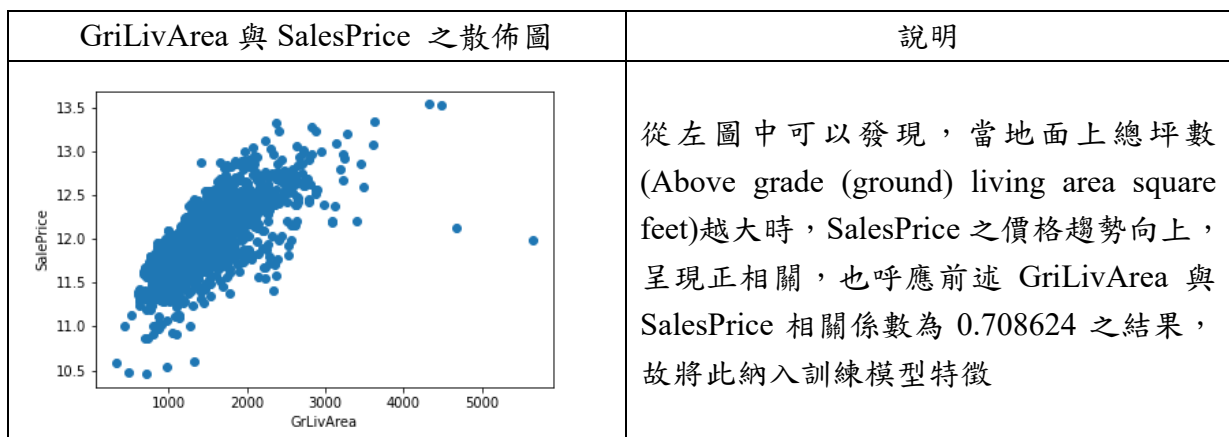
表 4 - SalePrice 與各類別變數之 t 分數表

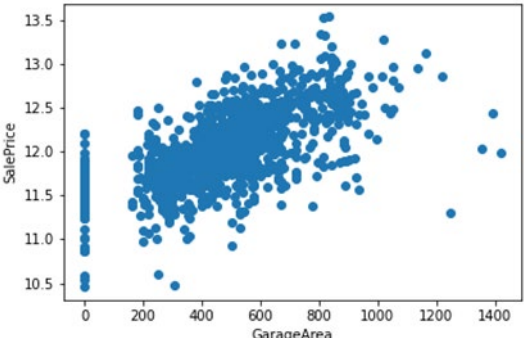
3. 資料視覺化

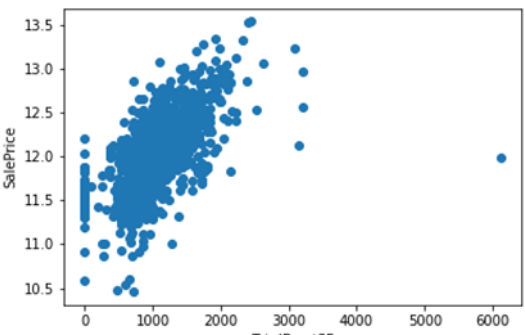
I. 連續型變數

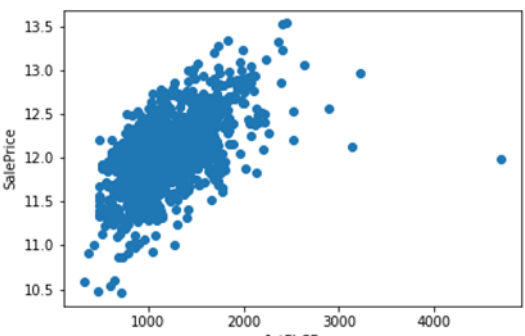
GrLivArea 代表的意思是地面上總坪數(Above grade (ground) living area square feet)、GarageArea 代表的意思是車庫坪數(Size of garage in square feet)、TotalBsmtSF 代表的意思是地下室坪數(Total square feet of basement area)、1stFlrSF 代表的意思是第一層樓的坪數(First Floor square feet)。

連續型資料我們採用散布圖的方式進行資料視覺化，X 軸為自變數，下列表格由上至下分別為 GrLivArea、GarageArea、TotalBsmtSF 及 1stFlrSF 與 SalePrice 之散佈圖，可以發現這些自變數和 Y 軸的目標變數都很明顯地呈現正相關，故我們將此 4 個變數皆納入訓練特徵。



GarageArea 與 SalesPrice 之散佈圖	說明
	<p>從左圖中可以發現，當車庫坪數(Size of garage in square feet) 越大時，SalesPrice 之價格趨勢向上，不過可以看見當房屋沒有車庫時，房價變異較大，其可能為導致 GarageArea 與 SalesPrice 相關係數只有 0.623431 之原因，不過仍呈現顯著正相關之結果，故將此納入訓練模型特徵</p>

TotalBsmtSF 與 SalesPrice 之散佈圖	說明
	<p>從左圖中可以發現，當地下室坪數(Total square feet of basement area)，越大時，SalesPrice 之價格趨勢向上，不過可以看見當房屋沒有地下室時，房價變異較大，其可能為導致 TotalBsmtSF 與 SalesPrice 相關係數只有 0.613581 之原因，不過仍呈現顯著正相關之結果，故將此納入訓練模型特徵</p>

1stFlrSF 與 SalesPrice 之散佈圖	說明
	<p>從左圖中可以發現，當第一層樓的坪數(First Floor square feet)越大時，SalesPrice 之價格趨勢向上，呈現正相關，也呼應前述 1stFlrSF 與 SalesPrice 相關係數為 0.605852 之結果，故納入訓練模型特徵</p>

II. 類別型變數

類別型變數我們多採用了長條圖來觀察各類別行變數與其平均 SalePrice 的關係，因變數眾多故我們依據表 5 之 significant 分數(*數越多代表影響目標變數越顯著)，從超過 2 顆*的變數中數挑選出 LotConfig、GarageQual、OverallQual、BsmtQual 來做視覺化分析。

LotConfig 代表的意思是土地面積的分布樣式(Lot Configuration)，類別選項有 Inside、FR2、Corner、CulDSac、FR3，從散佈圖我們可以發現 LotConfig 的各個類別似乎有離群值出現，且資料間存在變異，若將此變數作為訓練特徵，或許可使得模型學到更多東西。

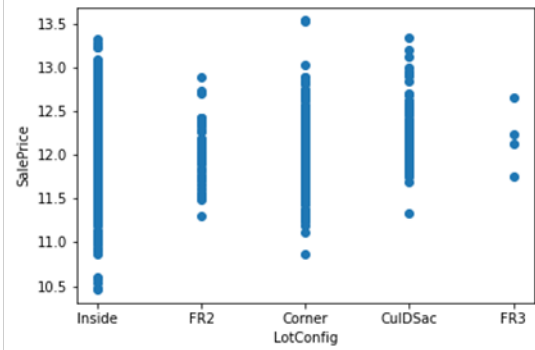
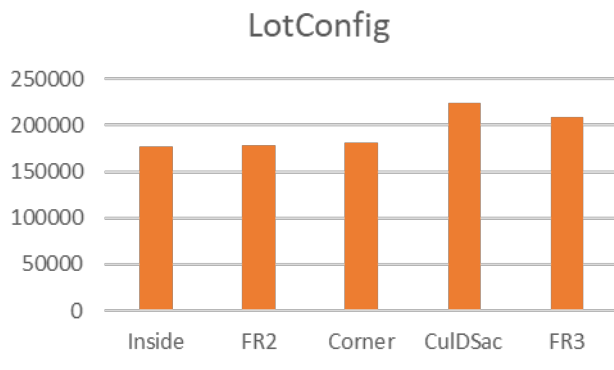


圖 2 - LotConfig vs. SalePrice

GarageQual 代表的意思是車庫品質(Garage Quality)，類別選項有 Ex、Fa、Gd、NA、Po、TA，從散布圖可以發現將 GarageQual 欄位包括進來進行預測，會有較理想的成效。

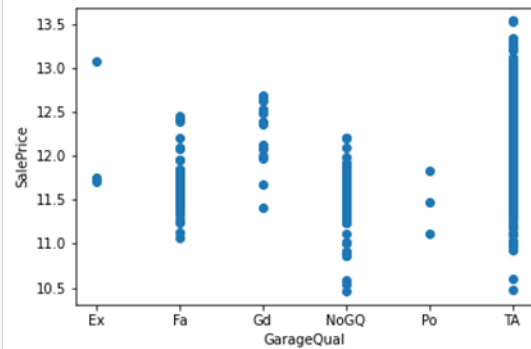
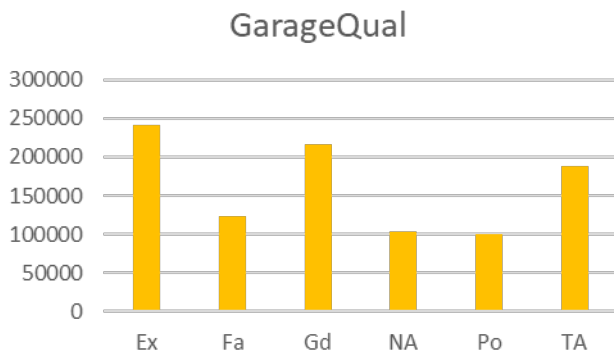


圖 3 - GarageQual vs. SalePrice

OverallQual 代表的意思是建材與施工品質(Overall material and finish quality)，類別選項有 1~10 分(離散)，1 最差，10 最好，從直條圖可以發現品質越好 SalePrice 平均越高，故將 OverallQualQual 欄位包括進來進行預測，會有較理想的成效。

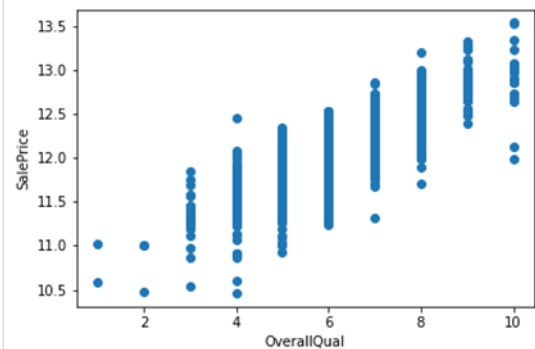
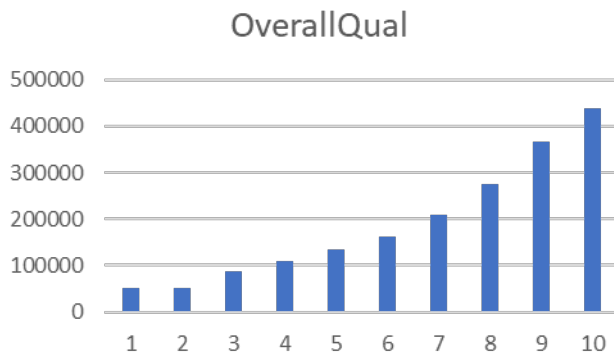


圖 4 - OverallQual vs. SalePrice

BsmtQual 代表的意思是地下室高度(Height of the basement)，類別選項有 Ex、Fa、Gd、NA、TA，從散布圖可以發現將 BsmtQual 欄位包括進來進行預測，會有較理想的成效。

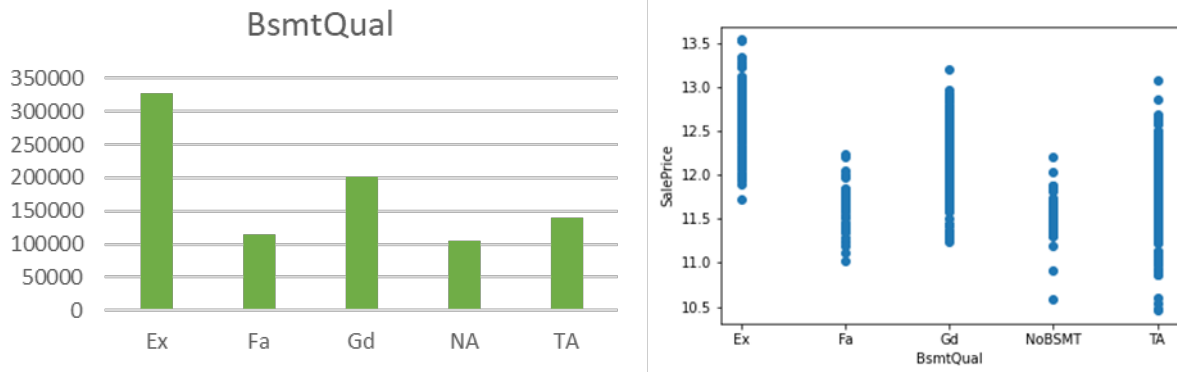


圖 5 - BsmtQual vs. SalePrice

4. 資料標準化

經上述資料觀察後，我們共挑選出 58 個變數作為我們的訓練特徵，並將原始資料分別經過補值、標準化後，才開始訓練模型，模型介紹及結果詳見第四、五章。

```
## Standardizing numeric features
numeric_features = features.loc[:,['LotFrontage', 'LotArea', 'GrLivArea', 'TotalSF',
                                   'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GarageArea']]
numeric_features_standardized = (numeric_features - numeric_features.mean())/numeric_features.std()

# MasVnrType NA in all. filling with most popular values
features['MasVnrType'] = features['MasVnrType'].fillna(features['MasVnrType'].mode()[0])

# BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2
# NA in all. NA means No basement
for col in ('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2'):
    features[col] = features[col].fillna('NoBSMT')

# TotalBsmtSF NA in pred. I suppose NA means 0
features['TotalBsmtSF'] = features['TotalBsmtSF'].fillna(0)

# Electrical NA in pred. filling with most popular values
features['Electrical'] = features['Electrical'].fillna(features['Electrical'].mode()[0])

# KitchenAbvGr to categorical
features['KitchenAbvGr'] = features['KitchenAbvGr'].astype(str)

# KitchenQual NA in pred. filling with most popular values
features['KitchenQual'] = features['KitchenQual'].fillna(features['KitchenQual'].mode()[0])

# FireplaceQu NA in all. NA means No Fireplace
features['FireplaceQu'] = features['FireplaceQu'].fillna('NoFP')

# GarageType, GarageFinish, GarageQual NA in all. NA means No Garage
for col in ('GarageType', 'GarageFinish', 'GarageQual'):
    features[col] = features[col].fillna('NoGRG')

# SaleType NA in pred. filling with most popular values
features['SaleType'] = features['SaleType'].fillna(features['SaleType'].mode()[0])

# Year and Month to categorical
features['YrSold'] = features['YrSold'].astype(str)
features['MoSold'] = features['MoSold'].astype(str)
```

```

features['GarageArea'] = features['GarageArea'].astype(float)
features['GarageArea'] = features['GarageArea'].fillna(0.0)

# GarageCars NA in pred. I suppose NA means 0
features['GarageCars'] = features['GarageCars'].fillna(0).astype(str)

# MSSubClass as str
features['MSSubClass'] = features['MSSubClass'].astype(str)

# MSZoning NA in pred. filling with most popular values
features['MSZoning'] = features['MSZoning'].fillna(features['MSZoning'].mode()[0])

# LotFrontage NA in all. I suppose NA means 0
features['LotFrontage'] = features['LotFrontage'].fillna(features['LotFrontage'].mean())

# Alley NA in all. NA means no access
features['Alley'] = features['Alley'].fillna('NOACCESS')

# Converting OverallCond to str
features.OverallCond = features.OverallCond.astype(str)

# MasVnrType NA in all. filling with most popular values
features['MasVnrType'] = features['MasVnrType'].fillna(features['MasVnrType'].mode()[0])

# Adding total sqfootage feature and removing Basement, 1st and 2nd floor features
features['TotalSF'] = features['TotalBsmtSF'] + features['1stFlrSF'] + features['2ndFlrSF']

```

圖 6 - 補值及標準化過程

四、 模型介紹

Layer (type)	Output Shape	Param #	
=====			
dense_257 (Dense)	(None, 256)	69632	

dense_258 (Dense)	(None, 256)	65792	

dense_259 (Dense)	(None, 256)	65792	

dense_260 (Dense)	(None, 128)	32896	

dense_261 (Dense)	(None, 1)	129	
=====			
Total params: 234,241			
Trainable params: 234,241			
Non-trainable params: 0			

```

model = Sequential()

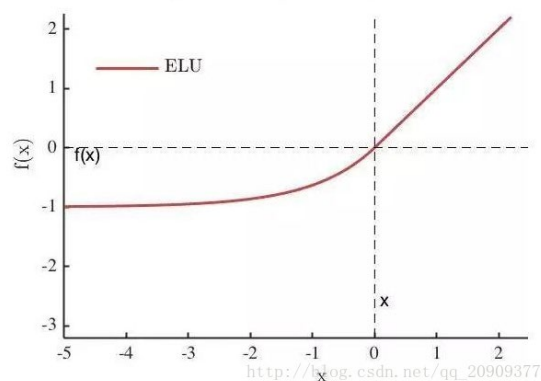
num_features = len(train_features[0])

model.add(Dense(256, input_dim=num_features,
                activation='elu',
                kernel_initializer='random_uniform',
                kernel_regularizer=regularizers.l2(0.01)
                ))
model.add(Dense(256, activation='elu',
                kernel_initializer='random_uniform',
                kernel_regularizer=regularizers.l2(0.01)
                ))
model.add(Dense(256, activation='elu',
                kernel_initializer='random_uniform',
                kernel_regularizer=regularizers.l2(0.01)
                ))
model.add(Dense(128, activation='elu',
                kernel_initializer='random_uniform',
                kernel_regularizer=regularizers.l2(0.01)
                ))
model.add(Dense(1, activation='elu',
                bias_regularizer=regularizers.l1(0.1),
                kernel_constraint=tf.keras.constraints.NonNeg()))

```

圖 7- 模型架構及參數圖

我們利用 python keras 來建構此深度學習預測模型，模型之架構及參數如圖 8，我們總共架設 5 個 dense 層，為避免 gradient vanishing problem，在研究各種 activation function 後，我們發現 elu 可以讓 optimizer 在參數在約等於零附近的區域能有更好的學習效率(詳見圖 9)，所以我們採用 elu 作為我們各層的 activation function。



The exponential linear unit (ELU) with $0 < \alpha$ is

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha (\exp(x) - 1) & \text{if } x \leq 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases}. \quad (15)$$

圖 8 - elu 數學函式圖

在多次實驗過程中，我們觀察到有 overfitting 及最後一層 bias 過大的現象，為了解決這些問題，我們前後各加入了 L2 kernel regularizer 防止模型過擬合，以及透過 L1 bias regularizer 降低最後一層 bias，讓神經元可正確學習 weights。模型訓練結果及超參數調整詳見第五章。

五、 成果與結論

本章分為兩小節，將分別介紹我們深度學習模型之超參數調整結果，第二節將說明該模型與傳統 Linear regression 及 SVR 等回歸模型之表現比較。

1. 超參數調整

在多次實驗中，我們分別調整了 epochs、Adam learning rate、hidden layer 神經元數及層數、kernel regularizer 等參數，以下調整前後的說明。

I. epochs 迭代次數 (500 vs. 1000)

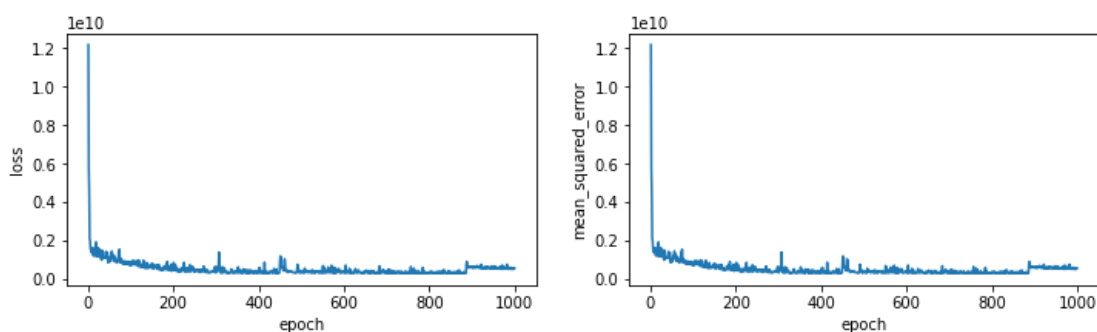


圖 9 - epochs=1000 之 training history

圖 10 為迭代次數等於 1000 的收斂過程，可發現該模型在迭代 400 次後，收斂已趨平緩，而超過 900 次後 loss 反倒增加，故最後我們以 500 作為最終的 epoch 設定。

II. Adam learning rate (0.01 vs. 0.001)

分別以 0.01 及 0.001 的學習率訓練出的模型，我們發現此二者除了在收斂速度上有差別外，對於學習成效沒有太大的影響，可能是因為此 model 的參數眾多，使得 local minimum 較不可能出現，故我們以 0.01 作為最終的 learning rate，讓模型可以快速收斂。

III. hidden layer 神經元數量 (128 vs. 256) / 層數 (2 vs. 4)

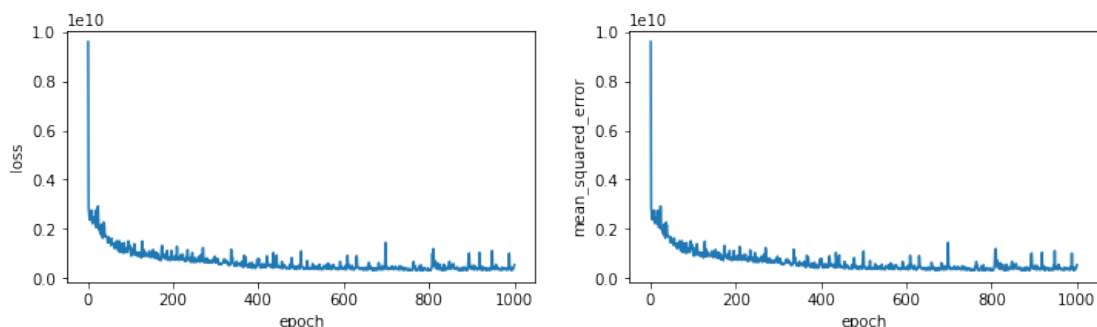


圖 10 - 兩層神經元數皆為 128 之 training history

我們試過僅有兩層 128 個節點的 Dense 層，其表現如圖 11，在 Kaggle 的表現已達到 0.14816，但我們加寬節點數至 256 後，在 Kaggle 的表現可以有更好的成效，達到 0.14121。而增加神經元數量後，我們又嘗試加深模型，最終加了一層寬度為 256 的 Dense 層，另一層寬度為 128 的 Dense 層，模型在 Kaggle 上的表現也有相對應的進步，達到 0.13656。

IV. kernel regularizer (0.01 vs. 0.001)

最後我們也試著調整 kernel regularizer 的參數，嘗試了 0.01 和 0.001，發現兩者對於訓練沒有太大的差異。可能是因為我們的 predict value 本身偏大，而 0.01 和 0.001 的 L2 regulation 的懲罰對於 Loss function 不足為奇。

2. 模型表現比較

因我們只有 1460 筆 training data，為防止欠擬合的問題產生，我們抓取其中 1387 筆資料作為訓練集，剩下 73 筆當作測試集，在經過 500 次 epoch 訓練及上節超參數調整後，訓練結果如圖 12 圖 12。由圖可知，模型收斂效果不錯，而我們接著比較模型預測值與真實值，可發現圖 13 之左圖預測值及真實值呈現一個線性的狀態，R-square 也達到 0.9434(越接近 1 越好)。而根據比賽的評比標準 Root Mean Squared Logarithmic Error，我們的 training set 表現也達到了 0.10(越低越好)，testing set 的表現也達到 0.13656。且透過圖 13 右圖可知，我們預測的值與實際值之間的差，呈常態分配，標準差為 17501.68，顯見我們的模型有不錯的預測效果。

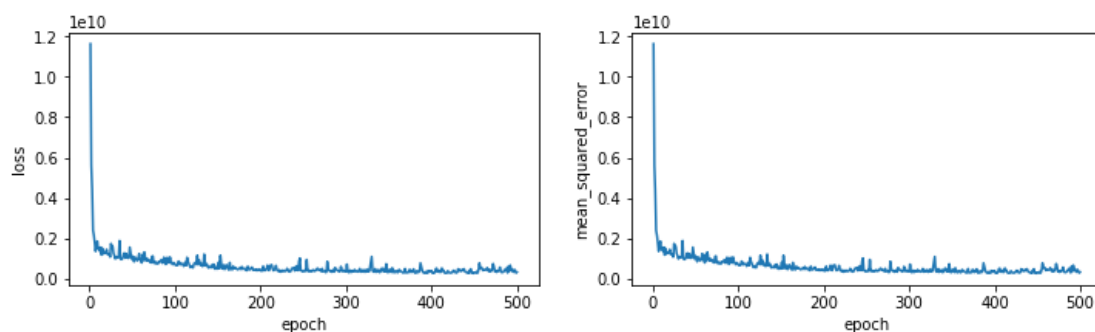


圖 11 - 最終版模型之 training history

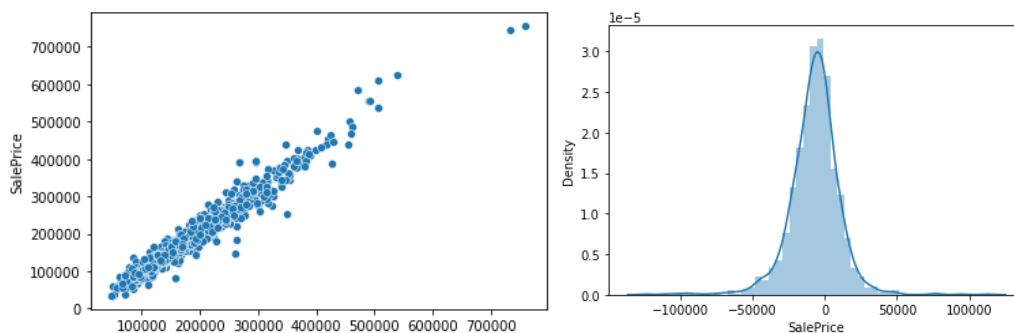


圖 12 - 預測值 vs. 實際值之視覺化結果

接著我們也用傳統的回歸方法 Linear regression 及 SVR 建構預測模型，模型架構及參數如圖 14、圖 15，兩者模型經訓練後之 R-square 分別為 0.897 及 0.94，可見 Linear regression 之擬合效果

不佳，而 SVR 的表現雖與深度學習模型相似，但我們將其預測結果丟至 Kaggle 上評測，得到了 0.15 的分數(該分數要越低越好)，顯見 SVR 有過擬合的現象產生，由下表 5 可見，也說明了我們建構的深度學習在此資料集中有最好的表現。

```
from sklearn.linear_model import LinearRegression

lm_model = LinearRegression(fit_intercept=True)

lm_model.fit(train_features_st, train_labels)
```

圖 13 - Linear Regression 模型架構

```
from sklearn.svm import SVR
clf = SVR(kernel='rbf', C=100000, gamma=0.01)
clf.fit(train_features_st, train_labels)

C=100000, cache_size=200, coef0=0.0, degree=3,
kernel='rbf', max_iter=-1, shrinking=True, tol=

y_test_raw = clf.predict(train_features_st)
y_test_raw
```

圖 14 - SVR 模型架構

訓練模型	R-square (越接近 1 越好)	RMSLE (越接近 0 越好)
Linear Regression	0.897	0.189
SVR(支援向量回歸)	0.94	0.15
深度學習預測模型	0.9434	0.10(training) / 0.13656(testing)

表 5 - 各訓練模型評估比較

總結以上，經由資料前處理(包含觀察缺失值、變數相關分析與資料視覺化)等手法找出特徵值作為訓練模型變數，並配合資料補值和標準化，最終挑選出 56 項變數作為訓練模型之因子，而透過 keras 深度學習模型建立，並透過實驗針對 epochs、Adam learning rate、hidden layer 神經元數及層數、kernel regularizer 等四種超參數進行調整，最終得出之訓練模型與其他回歸模型相比，有著最好的表現。

從本次的 Project 中，可以發現適當的篩選特徵變數及資料前處理手法對於後續的建立 Model 及辨識準確率極為重要，而對於深度學習各項超參數的理解與如何適當調整參數，更是報告帶給我們的學習重點，所以我們認為可以持續建立不同深度學習模型並依資料處後之特性來調整超參數或是運用於其他房價預測資料集(例如，台灣房地產資料等)來實作驗證，以此延伸本次研究內容與實際落地。

六、參考資料

1. <https://www.yongxi-stat.com/scale-stat/>
2. https://blog.csdn.net/qq_20909377/article/details/79133981