BÁO CÁO PROJECT 1

Nguyễn Đức Tú

I. Introduction

1. Competition

House Prices: Advanced Regression Techniques là một cuộc thi trên kaggle.com nhằm dự đoán giá nhà của vùng Ames,Iowa. Qua cuộc thi có thể củng cố kiến thức về feature engineering và các thuật toán machine learing.

2. Dataset

Dataset là bộ Ames Housing dataset được tổng hợp bởi Dean De Cock và được sử dụng cho mục đích học tập.

Trong bộ data gồm có 4 file:

- *train.csv*: tập dữ liệu dùng để train model
- *test.csv*: tập dữ liệu dùng để đánh giá model đã được train
- *data_descripton.txt*: mô tả chi tiết các thuộc tính dùng để dự đoán giá nhà
- *sample_submission.csv*: file dự đoán mẫu. Dựa theo format của file này để submit kết quả của mình.

Trong đó tập Train.csv có 81 features và test.csv có 80 features. 1 features còn thiếu trên tập test đó chính là *SalePrice*, thứ mà tôi sẽ dùng các kĩ thuật features engineering và machine learing để dự đoán.

3. Evaluation Metrics

Mục tiêu của cuộc thi này là dự đoán giá bán cho mỗi căn nhà. Với mỗi *ID* trong tập test cần phải dự đoán giá trị của *SalePrice*.

Metrics trong cuộc thi này là Root-Mean-Squared-Error (RMSE) giữa logarithm giá trị dự đoán và logarithm giá trị thực (Việc lấy logarithm làm cho việc dự đoán nhà đắt tiền hơn hay rẻ tiền hơn sẽ ảnh hưởng đến kết quả như nhau.

II. EDA

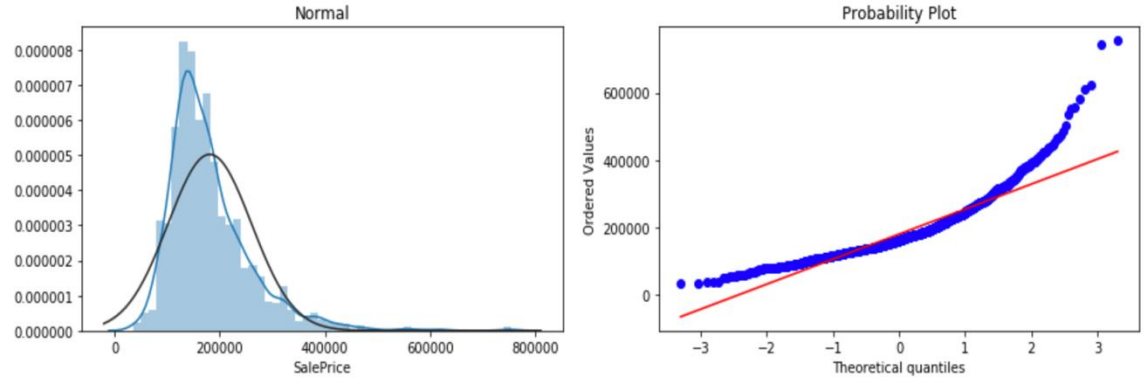
1. Load Data

Load 2 file *train.csv* và *test.csv* vào sau đó sẽ gộp 2 file này vào với nhau thành một file chung để thực hiện các feature engineering.

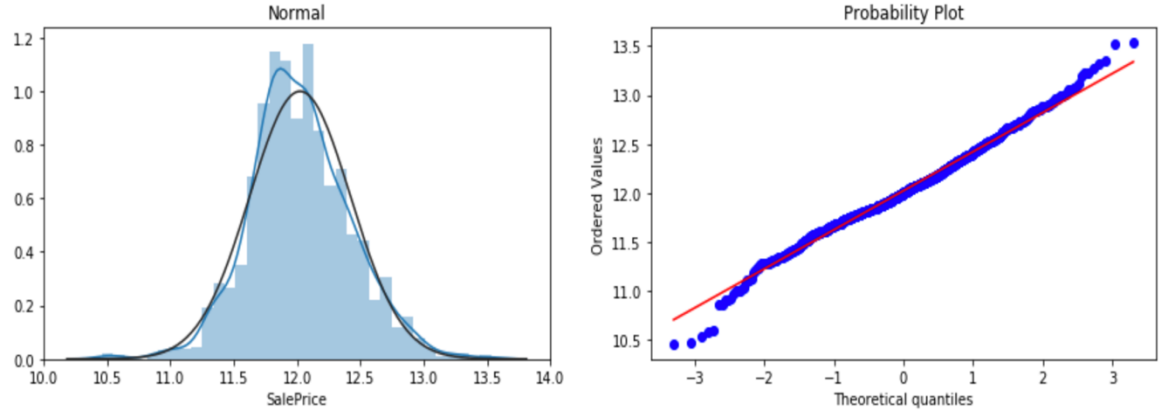
2. Analysing SalePrice

Sau khi đã load data xong tôi sẽ phân tích giá trị mà cần dự đoán đó là *SalePrice.*

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```



Có thể thấy *SalePrice* không tuân theo phân phối chuẩn nên tôi sẽ phải đưa các giá trị về dạng phân phối chuẩn. Ở đây tôi sẽ lấy *logarithm(SalePrice+1)* để đưa về dạng chuẩn. Việc cộng thêm số *1* ở đây là để tránh việc lấy *log(0).*



3. Handle Nan Values

Tiếp theo tôi sẽ xử lí các giá trị Nan ở trong data

	Total	Percent	Type
PoolQC	2909	0.996574	object
MiscFeature	2814	0.964029	object
Alley	2721	0.932169	object
Fence	2348	0.804385	object
FireplaceQu	1420	0.486468	object
LotFrontage	486	0.166495	float64
GarageFinish	159	0.054471	object
GarageQual	159	0.054471	object
GarageYrBlt	159	0.054471	float64
GarageCond	159	0.054471	object
GarageType	157	0.053786	object
BsmtCond	82	0.028092	object
BsmtExposure	82	0.028092	object
BsmtQual	81	0.027749	object
BsmtFinType2	80	0.027407	object
BsmtFinType1	79	0.027064	object

Thường thì những giá trị NaN là những missing values nhưng sau khi xem file *data_descripton.txt* tôi thấy có một số features giá trị Nan có nghĩa là không có feature đấy. Ví dụ: *GarageType* giá trị NaN có nghĩa là *no garage*. Với những features như vậy tôi sẽ thêm *None* vào các categorical features và *0* vào các numeric features. Với các features còn lại tôi sẽ điền các giá trị xuất hiện nhiều nhất đối với các categorical features. Riêng đối với *LotFrontage* tôi sẽ xử dụng *Ridge Regression* để điền các giá trị còn thiếu.

Sau khi xem xét tôi thấy một trường hợp đặc biệt đó là feature *Utilities*

```
AllPub    2916
NoSeWa       1
Name: Utilities, dtype: int64
```

Có thể thấy giá trị *NoSeWa* chỉ xuất hiện đúng một lần nên feature *Utilities* không có tác dụng gì cho việc dự đoán nên tôi sẽ xoá nó đi.

4. Add And Remove Some Features

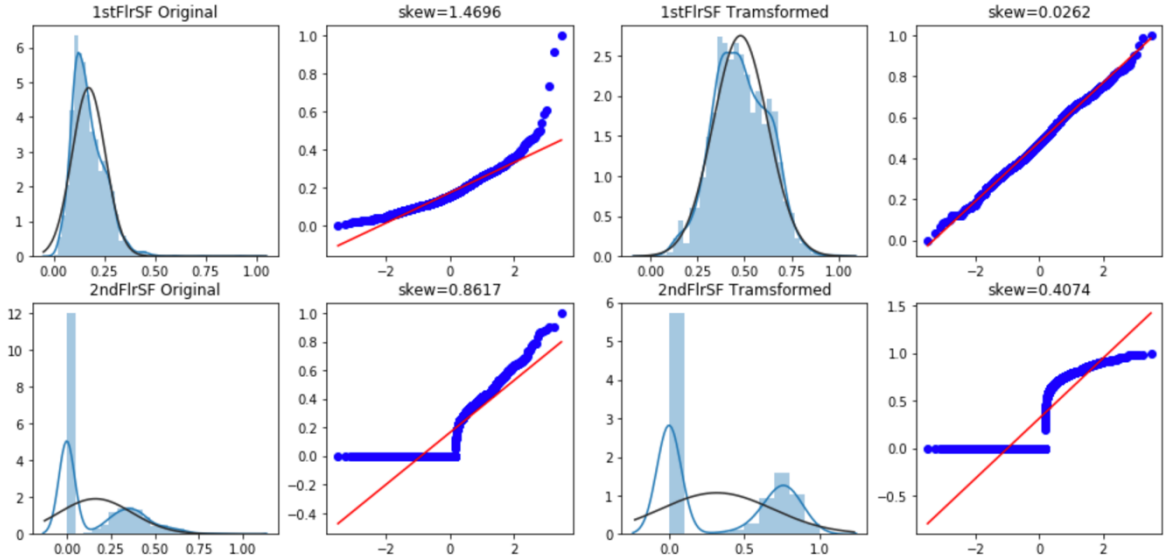
- Label Encoding: việc đầu tiên trong phần này là tôi sẽ chuyển một số các categorical feature về dạng ordinal vì thứ tự các giá trị của chúng có thể ảnh hưởng tới mô hình. Hầu hết đó là các giá trị về quality.
- Add new data: trong phần này tôi sẽ thêm một vài các features mới. Hầu hết các feature tôi thêm là các feature thể hiện sự có hay không của một feature khác. Ví dụ tôi sẽ thêm feature *hasPool* vì feature *PoolArea* có khá nhiều giá trị 0. Việc làm này sẽ làm cho mô hình ít ảnh hưởng bởi giá trị None đối với các categorical features và giá trị 0 với numeric features.
- Remove some data: Sau khi thêm các feature có hay không tôi sẽ xoá bớt các feature mà có quá nhiều giá trị None hoặc 0 và các feature mà theo thực nghiệm tôi thấy nó ảnh hưởng tới mô hình.

5. Identify Types Of Features

ở phần này tôi sẽ xác định xem đâu là các continuous, discrete, categorical feature.

6. Box Cox Tranformation With Continuous Features

Ở phần này tôi sẽ thực hiện chuyển đổi phân phối của một số các continuous features. Tôi dùng Box Cox Tranformation ở đây vì nó linh hoạt hơn Log Tranformation.

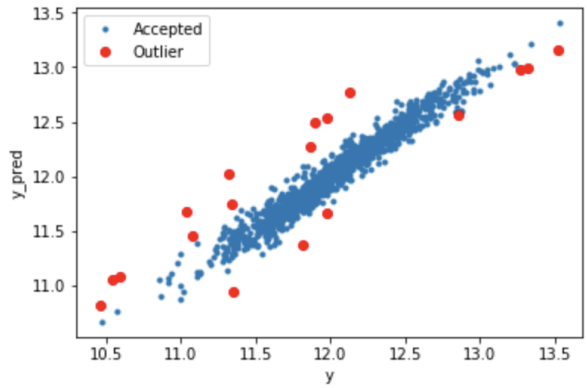


Có một số các feature mà việc tranformation không giúp ích gì nhiều thậm chí là còn tệ hơn (như ở trong hình là feature *2ndFlrSF*) bởi vì có quá nhiều giá trị 0 trong đó. Vì vậy tôi sẽ không thực hiện tranformation lên các feature mà việc này không giúp ích gì.

7. Find And Drop Outliers

Tôi sẽ sử dụng thuật toán *Ridge Regression* để tìm các outliers và xoá chúng đi

```
--------------------------------------
18 outliers:
[30, 88, 462, 495, 523, 588, 632, 681, 803, 825, 898, 968, 970, 1182, 1298, 1324, 1432, 1453]
```



III. Modeling

Các model được sử dụng là *ridge, lasso, elasticnet, xgboost,lightgbm, stacking*

1. Model

Tôi sẽ sử dụng GridCV để tìm ra các tham số phù hợp nhất với các model. Qua thức nghiệm tôi thấy việc cross validation chỉ nên thực hiện với các linear model là *ridge, lasso, elasticnet* còn các emsemble model là *xgboost và lightgbm* sẽ gây overfiting.

Vì các linear model khá nhạy cảm với nhiễu nên tôi sẽ scale data bằng *RobustScaler* trước sau đó mới đi qua model.

Đối với stack model tôi sẽ nhóm tất cả các model trên mà cuối cùng sẽ đi qua *ridge*.

Sau đó tôi sẽ fit tập train với tất cả các model và dự đoán trên tập test.

2. Blending Model

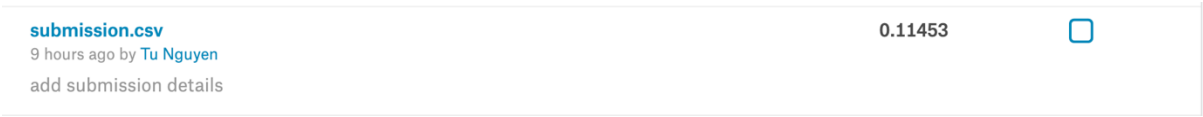
Ở phần này tôi sẽ kết hợp các dự đoán của tất cả các mô hình với nhau để được một dự đoán tổng quan nhất tránh overfitting.

3. Brutal Force và save model

Ở phần này tôi sẽ điều chỉnh một các giá trị ở các quantile nhất định để được kết quả tốt nhất và sau đó sẽ save lại các *SalePrice* đã dự đoán.

IV. Conclusion

Tôi được số điểm là 0.11453 top 11%



Đây là model đầu tiên của tôi nên kết quả vẫn chưa được tốt.