CCSEL1-18 Professional ELective

[House-Prices]

MARASIGAN, VEM AIENSI A.
3BSCS-1

Documentation and Leaderboard
January 2, 2024

Housing Price Prediction - Trial

0 Edit

Notebook Input Output Logs Comments (0) Settings

Add Tags

Table of Contents

This code is from the Step by Step Guide Notebook

1. Import Everything

In [1]:

```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import model_selection
```

In [2]:

```
#Import Data
df_train = pd.read_csv('../input/house-prices-advanced-regression-techniques/train.csv')
df_test = pd.read_csv('../input/house-prices-advanced-regression-techniques/test.csv')
```

Table of Contents

1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

2. Data Exploration

In [3]:

```
print('Train data shape : {}'.format(df_train.shape))
print('Test data shape : {}'.format(df_test.shape))
```

```
Train data shape : (1460, 81)
Test data shape : (1459, 80)
```

Note : Filling the null cells

In [4]:

```
null_columns = df_train.isnull().sum().sort_values(ascending=False)
sns.set(style="darkgrid")
plt.figure(figsize=(25,4))
sns.barplot(x=null_columns.index,y=null_columns)
plt.xticks(rotation=90)
plt.show
```

Out[4]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

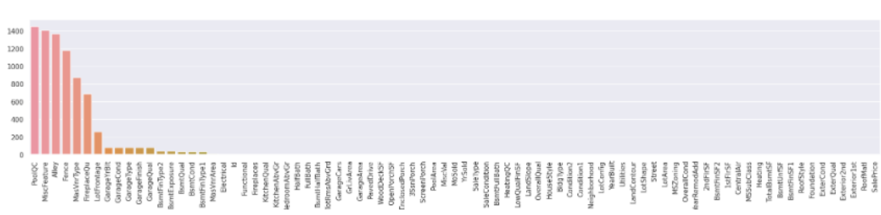


Table of Contents

1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

In [5]:

```
df_train.describe()
```

Out[5]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAd
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000

8 rows × 38 columns

MARASIGAN, VEM AIENSI A.
3BSCS-1

Documentation and Leaderboard
January 2, 2024

4. Feature Selection

Table of Contents


```
df_numeric = df_train.select_dtypes(include=['float64', 'int64'])
plt.figure(figsize=(15, 12))
sns.heatmap(df_numeric.corr(), annot=True, cmap='coolwarm', fmt=".1f")
plt.show()
```

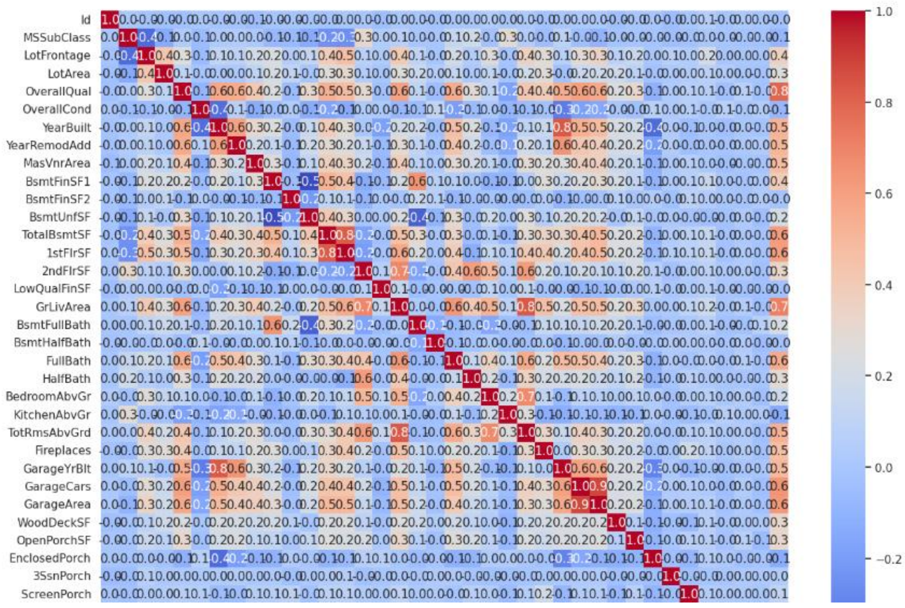
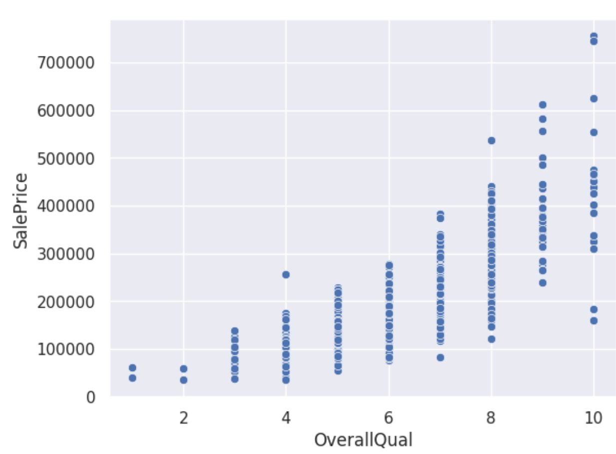


Table of Contents
1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

In [8]:

```
# First, Check for overall picture for important columns via scatterplots
final_features = ['OverallQual','LotArea','LotFrontage','BsmtFinSF1']
for item in final_features:
    plt.figure(figsize=(3, 3))
    fig, ax = plt.subplots()
    sns.scatterplot(x = df_train[item], y = df_train['SalePrice'],ax=ax)
    plt.show()
```

<Figure size 300x300 with 0 Axes>



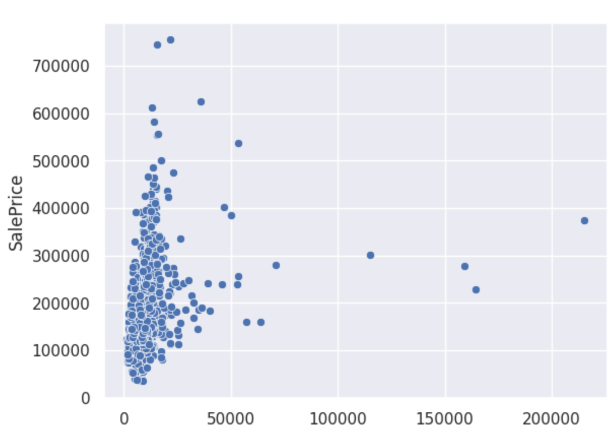


Table of Contents
1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

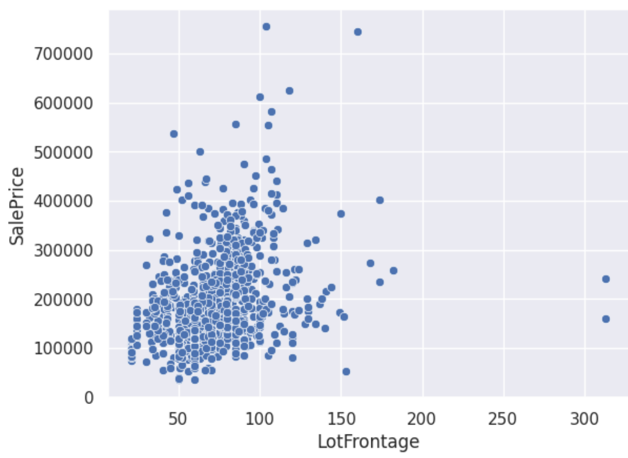


Table of Contents

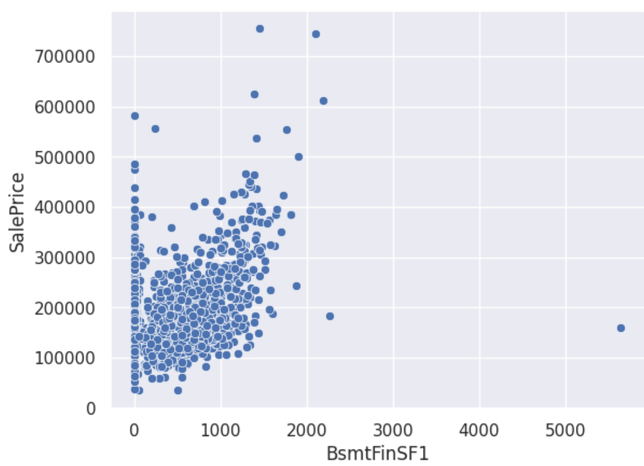


Table of Contents

1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

Note : After looking at all the graphs, it is clear that SalePrice > 500k , BsmtFinSF1 > 1700, LotFrontage > 150 and LotArea > 45k are anomalies. Let's remove it.

[9]:
```
df_train = df_train.loc[df_train['SalePrice'] < 500000]
df_train = df_train.loc[df_train['BsmtFinSF1'] < 1700]
df_train = df_train.loc[df_train['LotFrontage'] < 150]
df_train = df_train.loc[df_train['LotArea'] < 45000]
```

Table of Contents

1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

Note : For OverallQual, we cannot find outliers easily with this scatterplot. Let's try boxplot and try to choose good data. Boxplot method is very similar to Z-score outlier detection method.

In [10]:
```
sns.catplot(data=df_train,x='OverallQual', y='SalePrice', kind="boxen")
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x7dfda634b370>
```

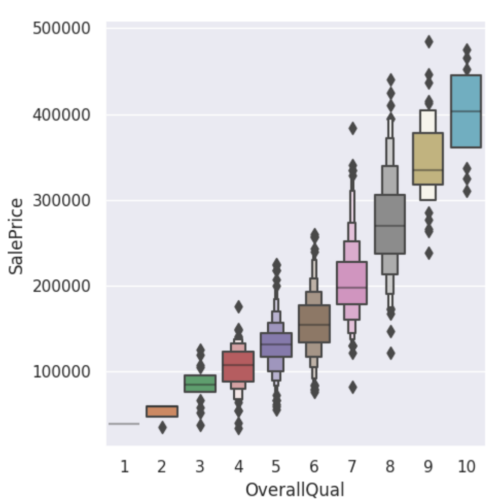


Table of Contents

6. Data preparation

In [11]:
```
# final_features = ['OverallQual','LotArea','LotFrontage','BsmtFinSF1']
final_features = list(set(df_train.columns) - set(['Id','SalePrice']))

X_train = df_train[final_features]
y_train = df_train['SalePrice']
X_test = df_test[final_features]
```

In [12]:
```
# Filter categorical columns
categorical_cols = [col_name for col_name in X_train.columns if X_train[col_name].dtype == "object"]
print('Categorical data columns: \n {}\n'.format(categorical_cols))

# Filter numerical columns
numerical_cols = [col_name for col_name in X_train.columns if X_train[col_name].dtype in ['int64',
'float64']]
print('Numerical data columns: \n {}'.format(numerical_cols))
```

```
Categorical data columns:
 ['BsmtCond', 'LandContour', 'RoofStyle', 'Foundation', 'BsmtExposure', 'Condition1', 'Heating',
'PavedDrive', 'FireplaceQu', 'BsmtQual', 'ExterCond', 'LotConfig', 'LotShape', 'Alley', 'Electri
cal', 'Utilities', 'HeatingQC', 'SaleType', 'PoolQC', 'BsmtFinType1', 'Neighborhood', 'GarageTyp
e', 'MiscFeature', 'ExterQual', 'CentralAir', 'Exterior2nd', 'MSZoning', 'GarageQual', 'HouseSty
le', 'Functional', 'KitchenQual', 'BldgType', 'MasVnrType', 'Exterior1st', 'SaleCondition', 'Gar
ageCond', 'Street', 'Fence', 'Condition2', 'GarageFinish', 'BsmtFinType2', 'LandSlope', 'RoofMat
l']
```

Table of Contents

1. Import Everything
2. Data Exploration
3. Feature Engineering
4. Feature Selection
5. Outlier Detection | Garbage...
6. Data preparation
7. Predict with ML | Training
8. Testing
9. Submission

7. Predict with ML | Training

In [14]:
```
GBR_model = GradientBoostingRegressor(random_state=0)
print(GBR_model.get_params())
```

```
{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': 'friedman_mse', 'init': None, 'learning_rate': 0.
1, 'loss': 'squared_error', 'max_depth': 3, 'max_features': None, 'max_leaf_nodes': None, 'min_i
mpurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_lea
f': 0.0, 'n_estimators': 100, 'n_iter_no_change': None, 'random_state': 0, 'subsample': 1.0, 'to
l': 0.0001, 'validation_fraction': 0.1, 'verbose': 0, 'warm_start': False}
```

In [15]:
```
GBR_training = Pipeline(steps=
                        [('preprocessing', preprocessor),
                         ('training', GBR_model)])
```

8. Testing

In [16]:
```
score = model_selection.cross_val_score(GBR_training, X_train, y_train ,cv=3)
score
```

Out[16]:
```
array([0.92477217, 0.90747973, 0.89584082])
```

9. Submission

In [17]:
```
GBR_training.fit(X_train, y_train)
predictions = GBR_training.predict(X_test)

submissions = pd.DataFrame({'Id': df_test['Id'], 'SalePrice': predictions})
submissions.to_csv('submission.csv', index=False)
```

In [18]:
```
# All of this does not make any sense to me yet
```

This link only points to a notebook a forked from the public code

https://www.kaggle.com/code/vemaiensi/housing-price-prediction-regression/edit

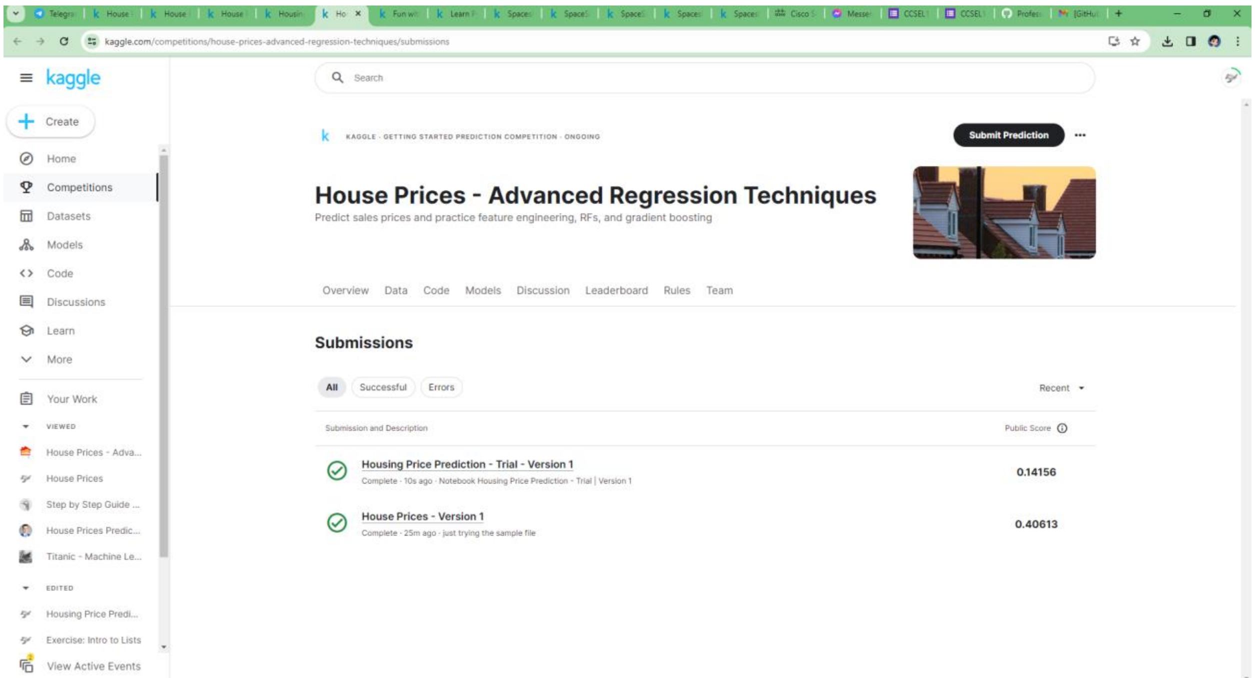
CCSEL1-18 Professional ELective

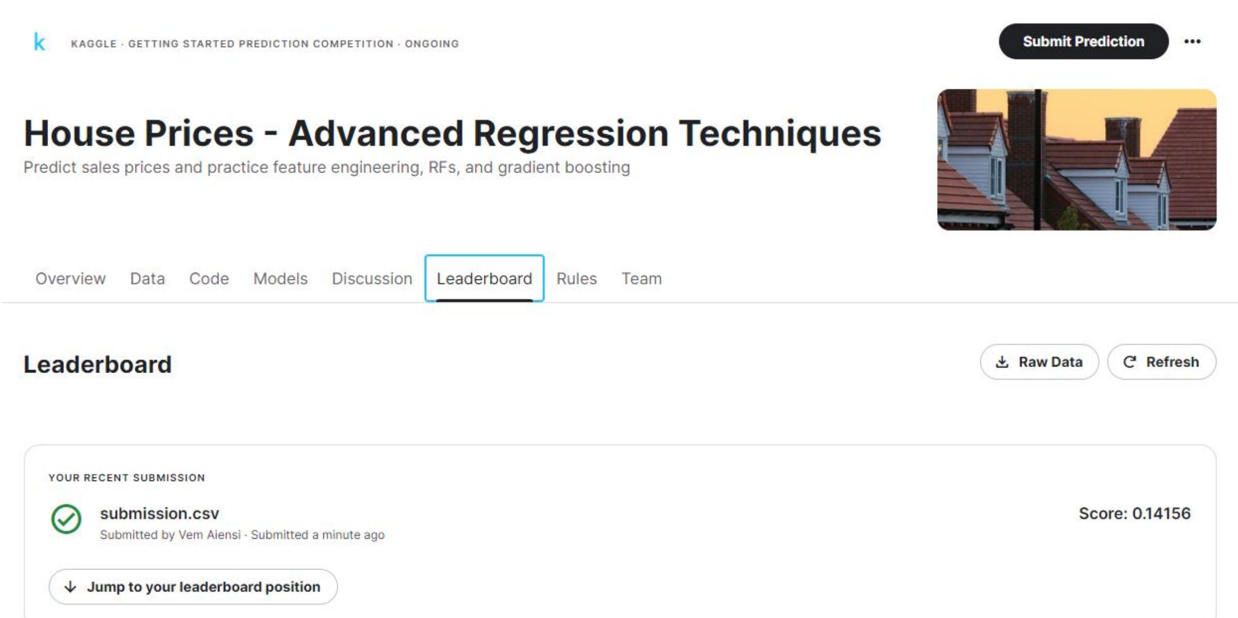
[House-Prices]

MARASIGAN, VEM AIENSI A.
3BSCS-1

Documentation and Leaderboard
January 2, 2024

SUBMISSION AND LEADERBORD SCORE





Files for this challenge

https://github.com/VemAiensi/Professional-Elective-Course/tree/main/Kaggle-Competiton/House-Prices

Other Competition

https://github.com/VemAiensi/Professional-Elective-Course/tree/main/Kaggle-Competiton