Capstone Proposal

December 19, 2016

1 Machine Learning Engineer Nanodegree

1.1 Capstone Proposal

1.1.1 House Prices

Hesham Shabana
December 18th, 2016

1.2 Proposal

1.2.1 Domain Background

House prices is a topic that always attract a lot of attention due to the daily demand and the wide range of interested party's investors, real estate agents, tax estimators, house owners, and house buyers with this the demand for a reliable model to assist a house price is needed. Traditionally house price prediction does not take into consideration the wide range of available parameters and it also assume independence between these parameters which does not hold in practice.

In Egypt buying a house is very important decision especially for young people who are looking to start a family and this decision becomes even harder with this increase in population, Egypt population estimated to be 93,383,574 with almost 2% growth in the last 4 years, and there is very little research in this area as well as a lack of data. Not to mention that people are following only one rule what is the square feet price in specific area? Therefore, using a machine learning algorithm to learn from the past purchasing history will help us to determine and predict the price of the house taking into consideration the attributes that matter the most and this will support any new buyer or a seller to set the right expectation.

Reference

- http://www.doc.ic.ac.uk/~mpd37/theses/2015_beng_aaron-ng.pdf
- https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US6609109.pdf
- https://researcharchive.lincoln.ac.nz/bitstream/handle/10182/5198/House_%20price_%20prediction.pdf
- http://link.springer.com/article/10.1007/s11146-007-9036-8

1.2.2 Problem Statement

The goal is to analyze historical sales of house prices features and through feature selection, feature engineering, machine learning finds the most relevant set of features that affect the price and which will allow us to perform an accurate prediction for new houses.

To avoid curse of dimensionality given the large number of features included in our database (80 feature) first we need to reduce our vector space by applying hill climbing to conduct feature selection, another proposed technique to reduce our feature space is to apply feature transformation by leveraging algorithms like PCA, ICA, Lasso regression. The goal here is to reduce our input space as much as possible to avoid overfitting.

1.2.3 Datasets and Inputs

This dataset describing the sales for houses in Ames, Iowa from 2006 to 2010 which contains 2930 observations and 80 variables, the variable distribution is 23 nominal, 23 ordinal, 14 discrete and 20 continuous.

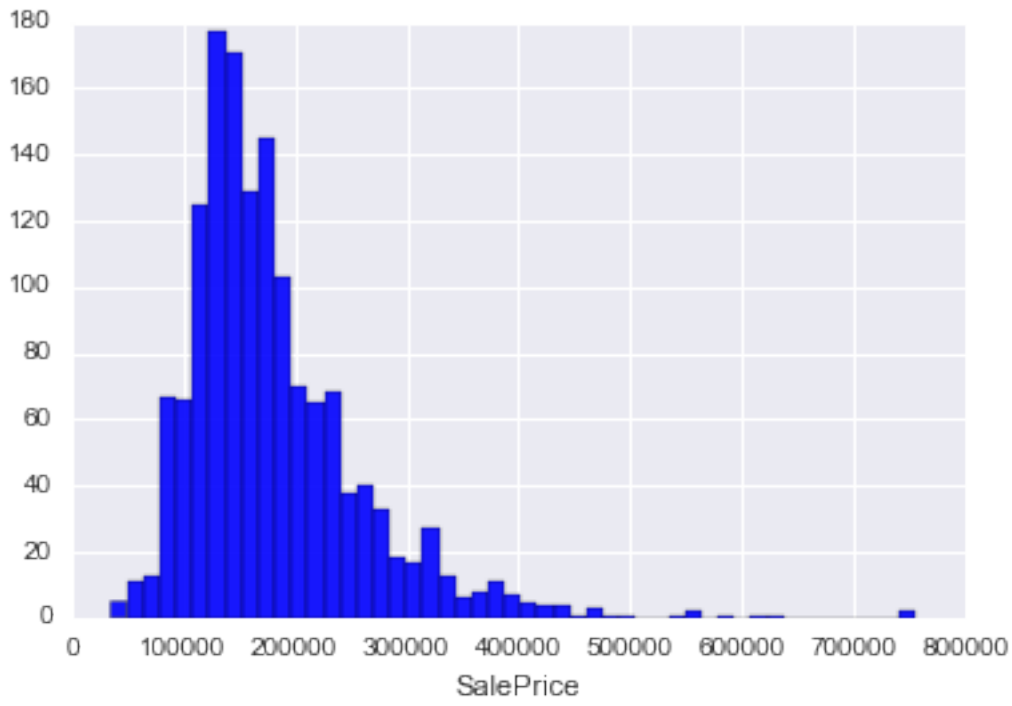
This dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

Statistics on the Housing Price variable:

```
In [1]: import warnings
        warnings.filterwarnings('ignore')

        import numpy as np
        import pandas as pd
        import seaborn as sns

        import matplotlib.pyplot as plt
        %matplotlib inline

In [2]: data = pd.read_csv('train.csv')

In [3]: data['SalePrice'].describe()

Out[3]: 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
```

Housing price distribution

```
In [4]: sns.distplot(data['SalePrice'], kde = False, color = 'b', hist_kws={'alpha': 0.9})

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x114f2e0d0>
```



As we can see that our distribution for the target variable is positive skew "skewed to the right", therefore, transform our data to be normally distribute may yield a better result, Box Cox transformations might be used.

$$T(Y) = (Y\lambda - 1)/\lambda$$

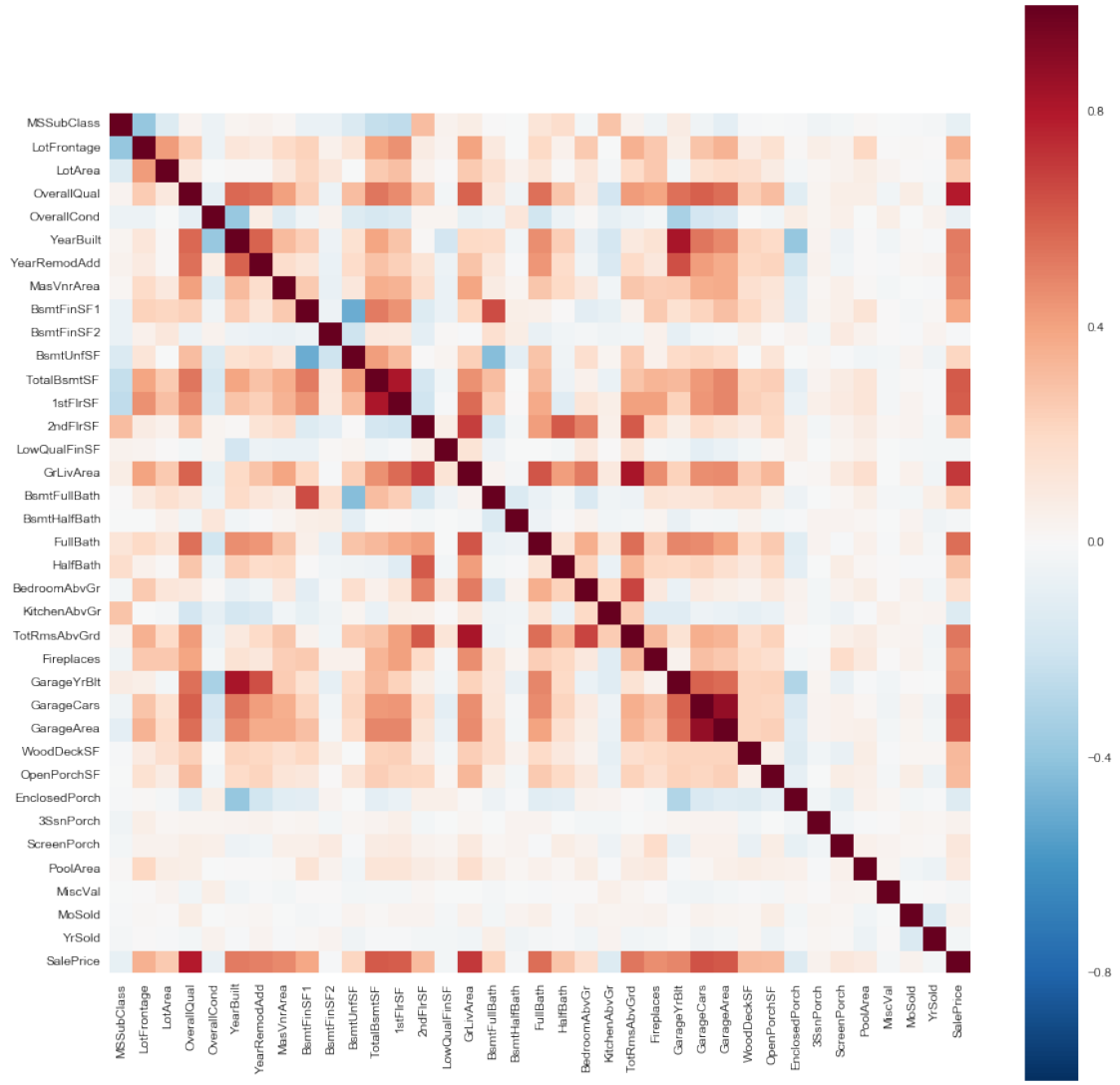
Box Cox transformations formula.

Data features

Numerical features The large number of continuous features (20) and discrete features (14) in this dataset give us various methods to combining and selecting the most relevant features.

```
In [5]: numerical_features_correlation = data.select_dtypes(include = ['float64', 'int64']).iloc[:, 1:]
        plt.figure(figsize=(15, 15))
        sns.heatmap(numerical_features_correlation, vmax=1, square=True)
```

```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1155e6710>
```



Categorical features This dataset has large number of categorical variables (23 nominal, 23 ordinal). They range from 2 to 28 classes with the smallest being STREET (gravel or paved) and the largest being NEIGHBORHOOD (areas within the Ames city limits). In addition, the nominal variables identify various types of dwellings, garages, materials, and environmental conditions while the ordinal variables typically rate various items within the property

```
In [6]: cat_features = data.select_dtypes(include=['object']).columns.values
```

```
In [7]: print cat_features
```

```
['MSZoning' 'Street' 'Alley' 'LotShape' 'LandContour' 'Utilities'
 'LotConfig' 'LandSlope' 'Neighborhood' 'Condition1' 'Condition2'
 'BldgType' 'HouseStyle' 'RoofStyle' 'RoofMatl' 'Exterior1st' 'Exterior2nd'
 'MasVnrType' 'ExterQual' 'ExterCond' 'Foundation' 'BsmtQual' 'BsmtCond'
 'BsmtExposure' 'BsmtFinType1' 'BsmtFinType2' 'Heating' 'HeatingQC'
 'CentralAir' 'Electrical' 'KitchenQual' 'Functional' 'FireplaceQu'
 'GarageType' 'GarageFinish' 'GarageQual' 'GarageCond' 'PavedDrive'
 'PoolQC' 'Fence' 'MiscFeature' 'SaleType' 'SaleCondition']
```

1.2.4 Solution Statement

A solution for this problem is to take the input features and make them suitable to be used by a machine learning algorithms, in addition, to do feature selection and engineering to enhance the performance of the model, at the end we should have a list of the most relevant features and an accurate model that could be used to predict new house prices.

1.2.5 Benchmark Model

The benchmark model is a linear regression model that predict the price based on the house size. In addition, calculating the Root Mean Squared Error between the logarithm of the predicted value and the logarithm of the observed sales price. (Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally.)

1.2.6 Evaluation Metrics

To assists the model performance, the below matrices will be used: * Bias: positive values indicate the model tends to overestimate price (on average) while negative values indicate the model tends to underestimate price. * Maximum Deviation: identifies the worst prediction made in the validation data set. * R^2 score: to measure the proportion of the variance that is predictable from another variable, which may help to remove a feature that has high correlation or explained by another feature

1.2.7 Project Design

To solve this problem the following steps shall be executed:

Data Preprocessing

- Feature Scaling VS. Feature Standardization
- Feature Scaling for continues variables so they can be compared on compound ground, scale down all the features between 0 and 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Feature Standardization: rescale features so they have the properties of normal distribution, as PCA may perform better after standardization.

$$z = \frac{x - \mu}{\sigma}$$

- Imputation of missing values
- One-Hot Encoding for categorical features
- Split the data to training and validation sets

Data Preprocessing Reference

Features

- Feature selection methods (Hill climbing, SelectPercentile, SelectKBest)
- Feature transformation (PCA algorithm, Lasso regression)
- Feature engineering

Training

- Train our model using the below algorithms:
- Logistic regression
- SVM
- Boosting
- Random forest

Evaluation

- Evaluate the model performance
- Apply our model to the testing data