**Machine Learning Engineer Nanodegree**

**Capstone Proposal**

Tony Chan   
May 25th, 2017

**Proposal**

**Domain Background**

This capstone will be focusing on Housing Price Prediction which is in financial industry. It is always interesting topic given it involves big investment for both developers and buyers. If there is a way to predict the housing price based on macro environment, it can help buyers to eliminate the information asymmetry which ultimately benefits buyers which most of them are using their 30 years saving just for one house.

One of the relevant research paper will be predicting London Housing Price from ICL.

<http://www.doc.ic.ac.uk/~mpd37/theses/2015_beng_aaron-ng.pdf>

**Problem Statement**

We aim to predict the housing price of Russia Price for potential buyers to judge whether they paid too much on their housing or not with considering macro & individual features. It is quantifiable given there is actual sold out price for many housings which we can compare with our prediction, it is measurable given there are lots of consistent features across time & housing such as PPI or living square feet which allows us to unify the information. It is also replicable given we can predict the future housing price and it will keep happening as there is always trade between housing.

**Problem Type**: Predicting Housing Price is a Regression Problem

**Evaluation Metrics**: RMSLE (Root Mean Squared Logarithmic Error)

**Potential Solution**: Regression Algorithm such as XGBRegression with Random Forest, Deep Learning with ANN

**Datasets and Inputs**

Basically, we have two input datasets should be considered for the project: i) Macro Environment dataset; ii) Individual Housing Information & Sold-out Price

Macro Environment – It is obvious that the macro economy will impact general housing price as if most of people earning money, they are willing to pay higher price for housing and vice versa. The data includes PPI, CPI, GDP etc features which help to predict value for general housing price across certain period/

|  |
| --- |
| **Examples of features regarding Macro Environment Dataset**  timestamp: Transaction timestamp  oil\_urals: Crude Oil Urals ($/bbl)  gdp\_quart\_growth: Real GDP growth  cpi: Inflation - Consumer Price Index Growth  ppi: Inflation - Producer Price index Growth  gdp\_deflator: Inflation - GDP deflator  balance\_trade\_growth: Trade balance (as a percentage of previous year)  usdrub: Ruble/USD exchange rate  eurrub: Ruble/EUR exchange rate  brent: London Brent ($/bbl)  net\_capital\_export: Net import / export of capital |

Individual Housing Information – However, it still has big variation for the price across different house given different size, rooms, and even the neighborhood. That is we will need information for each house such as full square feet, live square feet and most importantly, timestamp to link the housing with macro environment so that we can measure the impact.

|  |
| --- |
| **Examples of features regarding Individual Housing Dataset**  price\_doc: sale price (this is the target variable)  id: transaction id  timestamp: date of transaction  full\_sq: total area in square meters, including loggias, balconies and other non-residential areas  life\_sq: living area in square meters, excluding loggias, balconies and other non-residential areas  floor: for apartments, floor of the building  build\_year: year built  num\_room: number of living rooms  state: apartment condition  product\_type: owner-occupier purchase or investment  sub\_area: name of the district |

**Solution Statement**

Given it is a Regression Problem with more than 200 features, we will try in two different approach: i) Regression with XGBoost with Random Forest; ii) Deep Learning with Artificial Neural Network

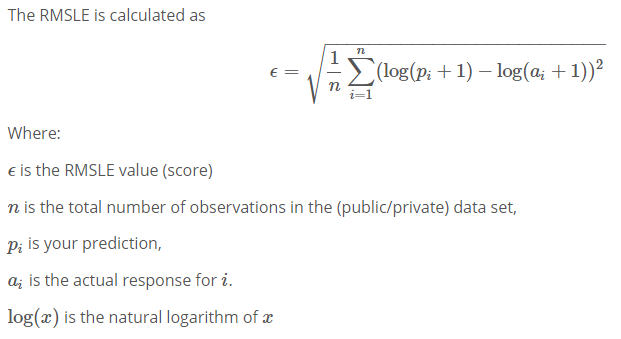
In this section, clearly describe a solution to the problem. The solution should be applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, describe the solution thoroughly such that it is clear that the solution is quantifiable (the solution can be expressed in mathematical or logical terms) , measurable (the solution can be measured by some metric and clearly observed), and replicable (the solution can be reproduced and occurs more than once).

**Benchmark Model**

We will benchmark with Simple Linear Regression Model. As we assume there is correlation between features in the dataset & the final pricing. We can check out whether it worth to have a higher computational model to have much more accurate prediction or not.

**Evaluation Metrics**

We will pick **RMSLE (Root Mean Squared Logarithmic Error)** as score for evaluation. There are usually 3 candidates for Regression Problem: i) MSE – Mean Square Error; ii) RMSE – Root Mean Square Error; iii) RMSLE – Root Mean Squared Logarithmic Error. Given Housing is a big investment to most of the buyers, it is needed to penalize the big diversion from Predicted Price to Actual Price so RMSLE is the best choice for evaluating the model



**Project Design**

1. Dataset Collection with python 3.6 environment

Use Pandas to import csv & visualize the data via statistical approach. Also pick some data points to get a sense of potential correlation

1. Feature Engineering on Dataset
2. Fill-up the missing information for all datapoint using Mean of the features
3. Feature scaling to put all features in same scale with lowering computational processing
4. Split Dataset into Train / Test dataset randomly
5. Model Fitting (Random Forest / Deep Learning – ANN)
6. Fitting 2 models to the train set & check out the initial accuracy within Train Set
7. Check out the result using test data to verify it is High Variance or High Bias
8. Feature Reduction (if Overfitting / Loading Time too slow for parameter tuning)
9. Select features that contributes up to 98% of variance using Principal Components Analysis
10. Repeat Step 3
11. Parameters Tuning with Cross Validation Check
12. Use gridSearch to test out parameters with k-fold validation (k = 10)
13. Get the best parameter and predict Test Result with best model