Friend team project proposal: House Pricing in King County, USA

Friend team, friend@chalearn.org
Walid Belrhalmia wbelrhalmia@gmail.com,
Paul-Hadrien Bourquin paulhadrien.bourquin@gmail.com,
Amine Biad biad718@gmail.com
Zizhao Li sharoncarl688@gmail.com,
Mo Yang lucasyeoh1992@gmail.com, Louis Trouche louisn.trouche@gmail.com.

February 19, 2018

GitHub:https://github.com/mogolola/houseprice_prediction.git
Starting-Kit:https://github.com/mogolola/houseprice_prediction/tree/master/starting_kit
WebSite:https://codalab.lri.fr/competitions/edit_competition/112
Video:https://youtu.be/z2Ughgz2PDk

Abstract

The purpose of this project is to create a challenge in which students will have to use the main tools of machine learning to make a prediction about house prices. Participants are provided with a dataset composed of many examples of house prices and the characteristics of these houses to train and test their models. In this report we describe the set up of the challenge and present some approaches for tackling this problem.

1 Background

"How much can I sell a house ?"
As a real estate agent, you will be trying to answer to this question by digging the data you have been given. In fact, this is a tricky question : you want to sell it at the higher price, but you also want to sell it and nobody will buy it if it's way overpriced. Your real estate agency in King County, a county of the state of Washington in the US, has all the records needed. You will look at the features of a lot of houses and the price they were sold at. We can guess that some properties are important, like a bigger house will probably be sold at a higher price than a smaller if they are both located in the same place, but perhaps there are some other factors not so obvious that might be taken into account when determining the price of a house. We want you to find those hidden factors that will allow you to make precise estimations of the right price of a house so we can make capital gains by buying undervalued houses and selling them thereafter.

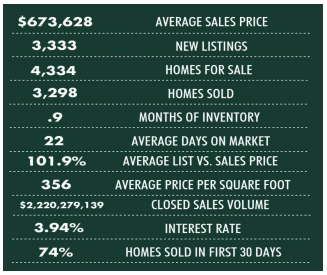

Figure 1: King country market analysis October 2017

According to a study made in October 2017 [1], the average sale price for King County homes is 673,628 dollars. In comparison 5 years ago the average sales price was 414,403 dollars. That is a 62 percent increase over 5 years ago.

2 Material and Method

2.1 Dataset

The dataset used is from Kaggle [2], House Sales in King County, a csv file containing 21613 observations. These sales were done between May 2014 and May 2015. Each line contains the price of the house sold, and it's characteristics : number of bedrooms, bathrooms, floor surface, number of floors, waterfronts, the location and some other values. The good news is that the file looks usable from the start ! We just have to parse the csv file and retrieve the values for each column.

We chose this dataset because it is a very good example of regression model : the price depends of multiple variables between the 19 given features. Some of them might be irrelevant and have no real impact on the price of the house, while some other will have massive impacts. The goal is to make the part of what data is relevant and what is not, and from the relevant part, be able to extrapolate the correlation between the values and the price associated with them.

2.2 Cheating Prevention

The definition of 'Data Leakage' in Kaggle is: The creation of unexpected additional information in the training data, allowing a model or machine learning algorithm to make unrealistically good predictions.[3] Moreover, if the original data is found and downloaded, contestants could cheat by over-fitting the model to test set. To prevent data leakage, we have first removed all column names, then shuffled all examples and columns. The data would be only numeric. Train set, validation set, and test set are divided in proportion of 0.6:0.2:0.2. Only Train data and their labels, Valid data is visible to contestants. The result of statistical analysis of the dataset is as below:

Type of predictable	Numeric
Nb of features	18
Nb of numeric features	10
Nb of categorical features	8
Nb of training examples	12967
Nb of valid examples	4322
Nb of test examples	4324
Missing data	Nah

Figure 2: Dataset statistic

2.3 Visualization and Exploration

After checking the quality of our dataset and verify that there is no missing values, we will start by visualizing the histogram of our target variable (price):

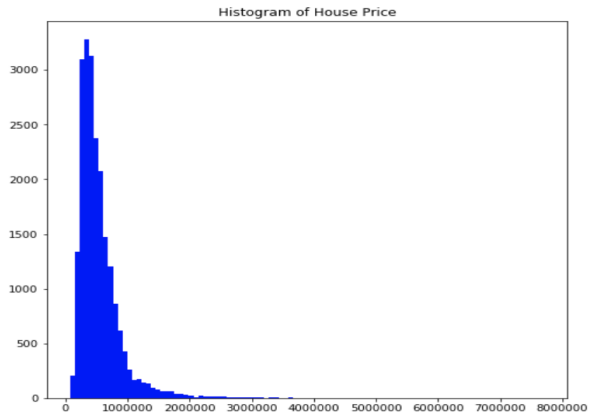

Figure 3: Histogram of house price

The following figure shows the distribution of the house prices. The plot is clearly shown that the distribution is skewed to the left. To be more precise, the skewness is 4.023, which is very high. A highly skewed data will affect the prediction result greatly. To improve the results, we apply log transformation on the house price to reduce the skewness, after applying the log the skewness is reduced to only 0.428.

Let's check the bivariate relationship between some variables with the housing price.

Figure 4: correlation of sqft Living with the house price

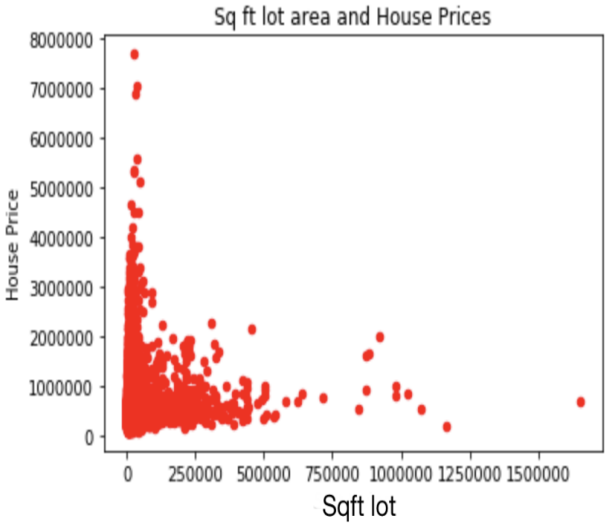

Figure 5: correlation of some features with the house price

Next we will check the seasonality of housing price (Note that here we will base our analysis only on the data for one year). In the following plot the y axis is representing the housing price on log scale, the color of the bubbles shows the number of sales. When the number of sales is high, the housing price is also high, which

is quite reasonable. But sometimes, the market might response a bit slow, that's why we can see in May of 2015, even though there are not many of sales, the price is still high. We can see there are clear seasonality based on the data of one year, so create some month indicators might be a good idea.

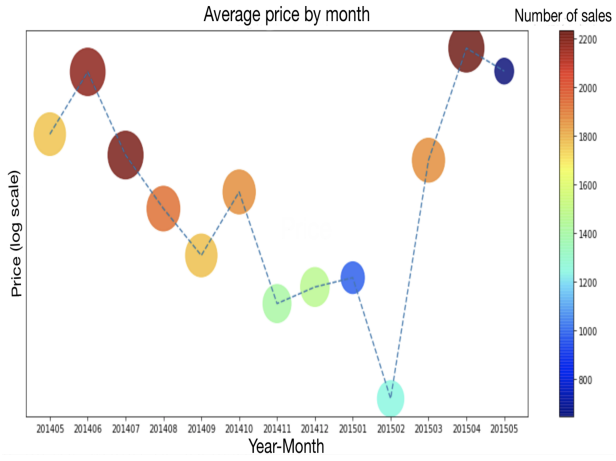

Figure 6: Average price per month

Another very important factor would be location. So let's check the housing price by latitude and longitude. As we can see from this graph, the northern area are generally more expensive than the southern. Most of the million-dollar houses (right stars in the graph) are around the hollow area in the north. That should be Lake Washington. Latitude and Longitude can be very good features if we want to build a model to predict the housing price. And they are on a more granular level than zip code. So it should provide more information than zip code. Another point to note from the graph is that the relationship between latitude/longitude and housing price is not linear, so linear model might not work very well. But tree based models and k-nearest neighbors algorithm should work better in this case.

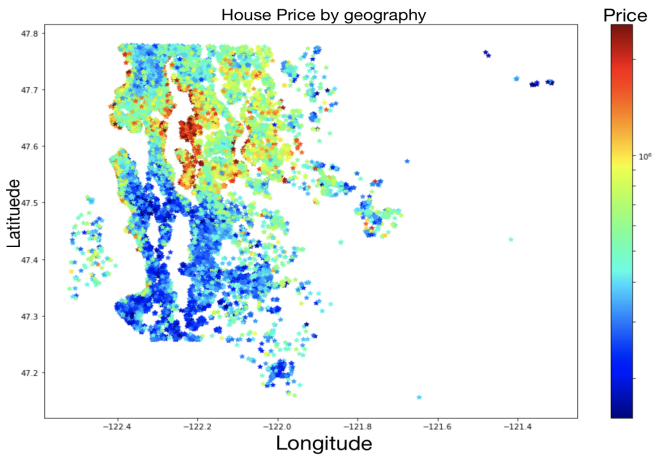

Figure 7: House price by geography

2.4 Preprocessing and feature extraction

In this part, our main goal is to find variables which have a strong correlation with the house prices. To do so, we must identify two types of variables, Categorical variables and continuous variables.

Correlation for categorical variables:

By viewing the possible values of our variables , we can identify 7 categorical variables; waterfront, bedrooms, bathrooms, floors, view, condition, grade. Then, we computed the correlation of this variables with the house prices, the using of correlation [4] is justified since our categorical variables have binary or discrete values :

```
The Correlation of waterfront with price is PointbiserialrResult(correlation=0.26636943403060209, pvalue=0.0)
The Correlation of bedrooms with price is (0.30834959814563828, 0.0)
The Correlation of bathrooms with price is (0.52513750541396187, 0.0)
The Correlation of floors with price is (0.25679388755071841, 1.5810100666919889e-322)
The Correlation of view with price is (0.39729348829450428, 0.0)
The Correlation of condition with price is (0.036361789128997554, 8.9356540623440942e-08)
The Correlation of grade with price is (0.66743425602023709, 0.0)
```

Figure 8: The correlation of categorical variables with house prices

As results shows, the top 3 categorical variables that have the highest correlation with house prices are: grade (0.66), bathrooms (0.52), view (0.39)

Correlation for continuous variables:

we will use the correlation heatmap in order to analyze the correlation of continuous variables, the top 3 continuous variables are: sqft_living (0.7), sqft_above (0.61), sqft_living15 (0.59)

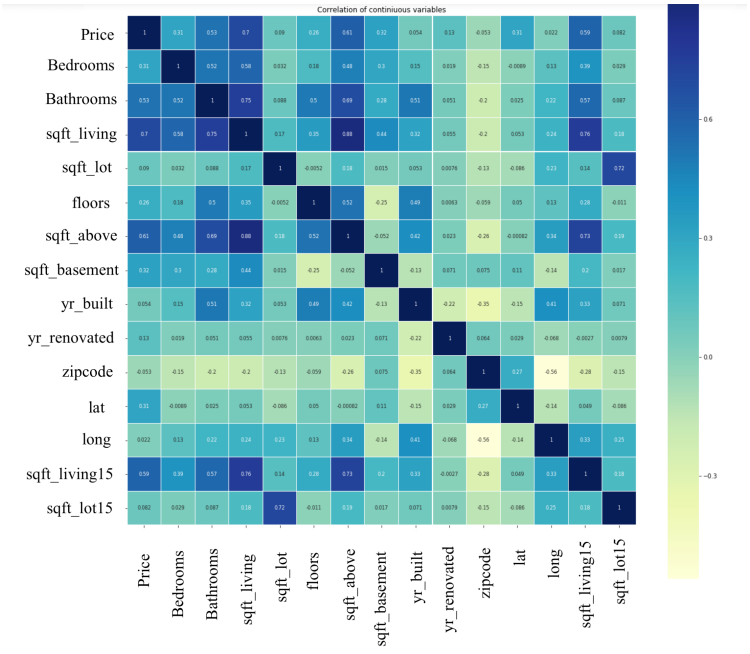

Figure 9: The correlation heatmap for continuous variables

Conclusion, from above we can determine the top 5 most correlated and important features: sqft_living=0.7, grade=0.66, sqft_above=0.61, sqft_living15=0.59, bathrooms=0.52.

Checking Data Leakage : all correlated variables with our target (price) are legitimates, and since we don't have any missing values we can conclude that we don't suffer from data leakage.

2.5 Advanced features ranking using RFE

Datasets used to train classification and regression algorithms are high dimensional in naturethis means that they contain many features or attributes. However, Not all features contribute to the prediction variable. Removing features of low importance can improve accuracy, and reduce both model complexity and overfitting. Training time can also be reduced for very large datasets. For our project, we will use Recursive Feature Elimination (RFE) with Scikit Learn.

Recursive Feature Elimination or **RFE** uses a model (eg. linear Regression or SVM) to select either the best or worst-performing feature, and then excludes this feature. The whole process is then iterated until all features in the dataset are used up (or up to a user-defined limit). Sklearn conveniently possesses a RFE function via the sklearn.feature_selection call and we will use this along with a simple linear regression model for our ranking

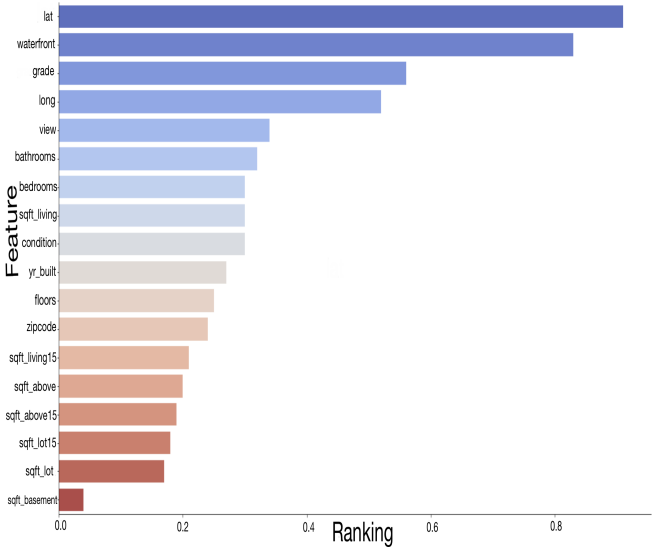

Figure 10: Features Ranking RFE

Well as you can see from our feature ranking, the top 3 features are 'lat', 'waterfront' and 'grade'. The bottom 3 are 'sqft_lot15', 'sqft_lot' and 'sqft_basement'. We will be using this feature selection rather than the one giving by the correlation matrix since it is more reliable and use advanced techniques.

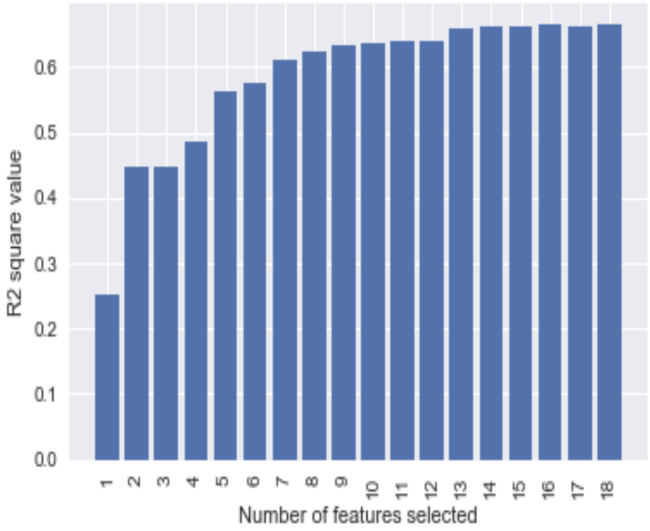

Figure 11: R-squared according to the number of features

2.6 Feature Extraction: Principal Component Analysis (PCA)

The goal of Principal Component Analysis (PCA) is reducing the dimensionality of data that contains variables with correlation between each other, while maintaining the variation of the data, to the maximum extent (Deyzre). The variables that have lower variance, which are not spread out a lot, will be projected to a lower dimension. Then, the model will be trained on this transformed data.

The following figures show how much variance the principal components explained in ratio and in plot. The first three principal components have about 50% variance explained. Principal components four to nine have about 35% variance explained.

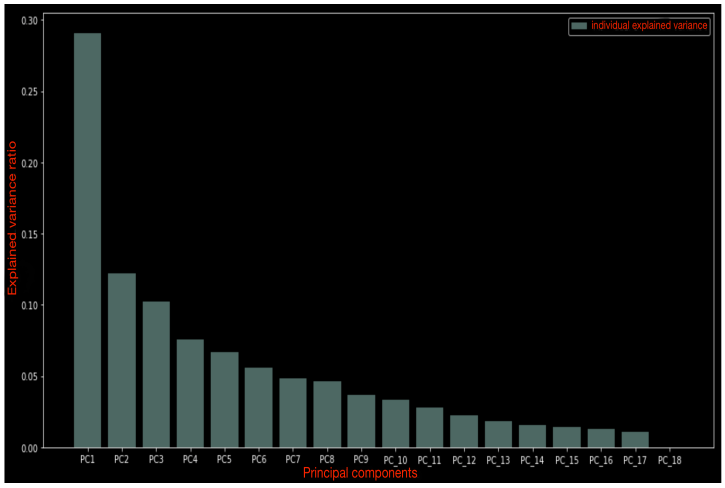

Figure 12: principal components explained in percentage

So, we will choose the number of principal components that explain most of the variation. we will pick sixteen principal components since they explain over 98% of the variation.

2.7 Evaluation and metrics

In order, the measure the performance of the model, we will be using the R-Squared metric [5], This is probably the most commonly used statistics and allows us to understand the percentage of variance in the target variable explained by the model. Its easier to interpret than other metrics because its bounded between 0 and 1(Higher is better).

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R-squared = Explained variation / Total variation

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

Plotting fitted values by observed values graphically illustrates different R-squared values for regression models:

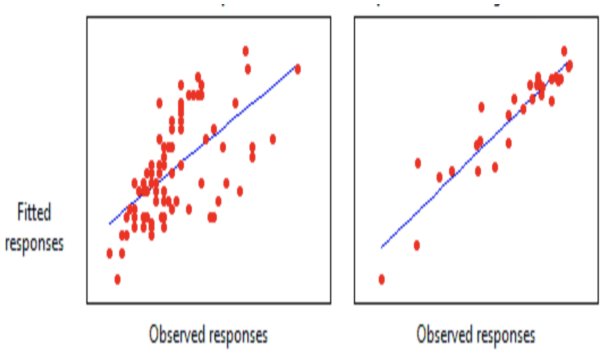

Figure 13: Plots of observed responses versus fitted responses for two regression models

The regression model on the left accounts for 38.0% of the variance while the one on the right accounts for 87.4%. The more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line. Theoretically, if a model could explain 100% of the variance, the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line.

3 Results:

So, here basically we will train a linear Regression[6] Model, we will train our model on our 60% train data and test it on our validation set. So we got a result of:0.83+-0.02 for the R-square. Very good score, we will try to find another model that give much higher results so we compared multiples regression model and test their performances on our data.

Model	R2
Decision tree	0.790989
Random forest	0.868716
Knn	0.456395
MLPR	0.571
GradientBoost	0.884627

Figure 14: Results on some predictive algorithms

Typically, gradient boosted decision trees(XgBoost [7]) give the best results with an r2 error of 0.88. In order to find this value of the r2 error with the XgBoostRegressor, we needed to find the best values for the parameters. So we did a GridSearch [8] and foud that the best values were : $colsample\_bytree$: $0.7, learning\_rate$: $0.1, min\_child\_weight$: $10, n\_estimators$: $500, subsample$: $0.9, objective$: $reg : linear, max\_depth : 7, gamma : 0, reg\_alpha : 1$

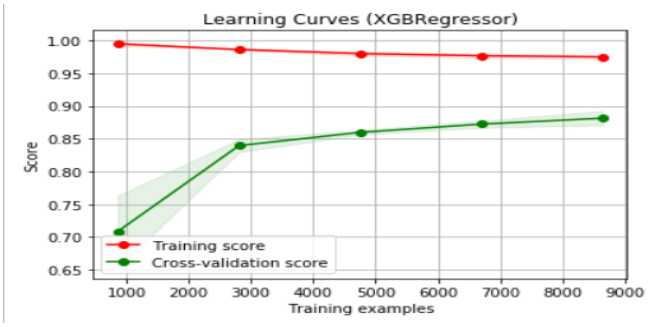

Figure 15: Learning curve [9] of the XGBoost

We see on that figures that we obtain good results with the number of data we have in our training set so we don't need anymore data in our training set. Also We checked whether the order of our variables does not inform about the target. we done that by shuffling all our examples in random order to make sure(avoiding Data Leakage).

3.1 SVM

We have also tested SVM [10] for the tasks. We have used PCA [11] for features selection. The PCA feature dimension is 16. First we have used GridSearch to tune the model. We have used a rather small dataset to tune the model due to the complexity of algorithm. We have chosen 10% training data which is randomly distributed from the dataset and have separetely tuned SVM with linear kernel and gaussian kernel. For gaussian Kernel, we were tuning two hyperparameters: C and gamma. For linear kernel, we were tuning hyperparameter C. Each

search in the grid was validated by 5-folds cross validation. Here is the result of tuning:

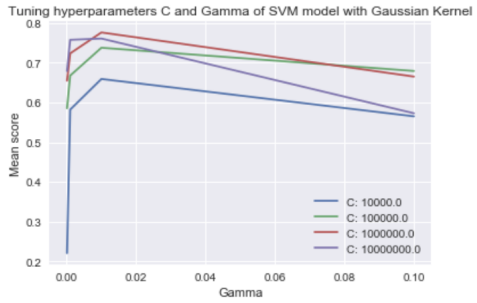

Figure 16: Results of GridSearch for SVM with gaussian kernel

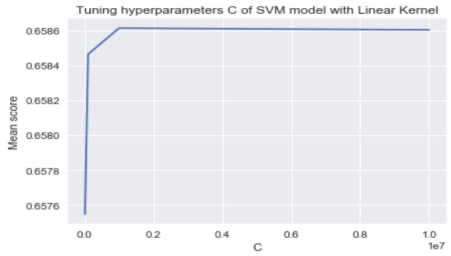

Figure 17: Results of GridSearch for SVM with linear kernel

From the result above we have got the optimum hyper parameters for SVM: {'kernel':'rbf', 'C':1e7, 'gamma':1e-2 }. Then we have drawn the Learning Curves as a function of number of training examples. The result is as below:

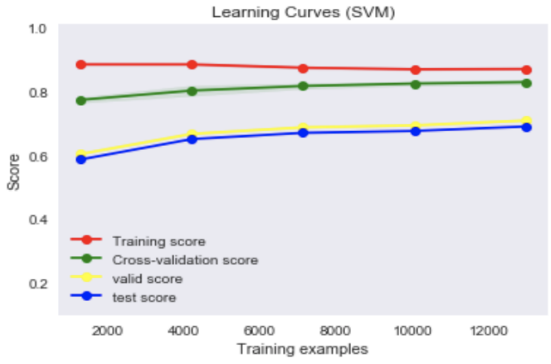

Figure 18: The learning curve of SVM

It took a really long time to run SVM on the whole training set. The time complexity of SVM is $O(mn^2)$ (m:number of features, n: number of examples). When we have large dataset, the space complexity is also very high, even larger than cache. Thus, the fit time complexity is even higher than quadratic with the number of samples which makes it hard to scale to large dataset. From the result we can also observe of its good generalization to rather small dataset.

3.2 SGD

The following figure show the learning curves of the SGDRegressor [12] model :

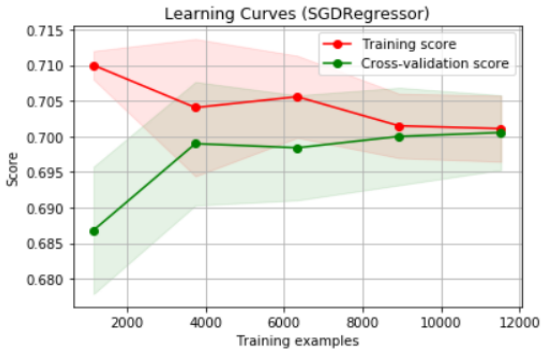

Figure 19: SGD

We see that the results of the SGDRegressor are cleary inferior to those of the XGBoost so we will not use this model to solve our problem.

4 Conclusion

We created a challenge based on a regression problem which aims to predict house prices in the north west area of the United States. We worked on features extraction to show which ones are the significant ones. Moreover, we reduced the dimensionality of the problem using PCA and we selected 16 of them. Thereafter, we presented the scores of some predictive algorithms and run some tests on hyperparameters for SVM and SGD. We noticed that gradient boosted decision (XgBoost) give the best results for this dataset, and the score that we obtained is the highest among other competitors which have posted their solution about this dataset on Kaggle.

Reference

[1] King County Housing Market Report 2017. `http://www.themadronagroup.com/king-county-real-estate-market-analysis`.

[2] Dataset: House Sales in King County, USA. `https://www.kaggle.com/harlfoxem/housesalesprediction`.

[3] Data Leakage Definition. `https://www.kaggle.com/wiki/Leakage`.

[4] Data, Covariance, and Correlation Matrix. `http://users.stat.umn.edu/~helwig/notes/datamat-Notes.pdf`.

[5] R-squared Definition. http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.

[6] Linear Regression Example. http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html.

[7] XGBoost librairy for Python. http://xgboost.readthedocs.io/en/latest.

[8] Exhaustive Grid Search. http://scikit-learn.org/stable/modules/grid_search.html.

[9] Learning curve. https://en.wikipedia.org/wiki/Learning_curve.

[10] Support vector machines (SVMs). http://scikit-learn.org/stable/modules/svm.html.

[11] PCA Explained. https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf.

[12] Stochastic Gradient Descent (SGD). http://scikit-learn.org/stable/modules/sgd.html.