Estimating House Price: Regression Techniques

Deepshi Mediratta (300086013)

Faculty of Engineering, University Of Ottawa, Ottawa, Canada
dmedi018@uottawa.ca

Abstract. This paper aims at training machine learning models to predict house price. We use the Melbourne Housing Market dataset for our analysis. The data is composed of samples of properties built between 1196 and 2016. The dataset is processed and analyzed using Python and its libraries such as sklearn, numpy, pandas etc. The relationship between the features are analyzed using heat maps and select K best algorithm to select the features. The price predictions are made based on features such as number of rooms, building area, land size, location etc. We train and compare various regression models to come up with the best results. Models discussed include linear model, tree-based, distance based, rule-based and ensembles.

Keywords: House price prediction, regression model, real estate

1 Introduction

House prices are deeply connected to the economy of a country. Whenever there is a fluctuation in the property price trend, it tends to affect various factors such as consumer spending, property development, construction employment etc. Australian housing market has witnessed recurrent inconsistency due to economic shift and market changes. The variations in house prices and the other factors that influence the value of a property, makes it difficult for home buyers and sellers to assess the property and come up with an amount which is suitable.

Real estate agents typically value a property based on a few comparable properties. The unreliability of the housing market and the current method of establishing the selling price for a house, compels us to come up with a better solution. Hence, a predictive study for House price forecasting has gained popularity in recent years.

This paper focuses on the Melbourne Housing Market dataset and regression models that can calculate the price of a property based on it's features such as number of rooms, area, year built, car parking, house type, land size, building area

etc. Given the diversity in the features of the samples, it is challenging to build a pattern and train a model that accurately predicts the price. The dataset is processed and cleaned before feeding into the model. Feature selection is based on heat maps and select k best algorithm.

2 Case Study- Melbourne Housing Market

2.1 Data Description

The dataset is composed of the prices and features of actual residential houses that were built between 1196 and 2016, scraped from public results posted weekly on [1]. This dataset consists of 34,857 house samples with 20 features and property's price as the target value.

Table 1. Melbourne Housing Market Feature description

Feature	Feature Description
Suburb	Suburb
Address	Address
Rooms	Number of rooms
Type	br - bedroom(s); h - house, cottage, villa, semi, terrace; u - unit, duplex; t - townhouse; dev site - development site; o res - other residential
Price	Price in Australian dollars
Method	S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN - sold not disclosed; NB - no bid; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS - sold after auction price not disclosed. N/A - price or highest bid not available.
SellerG	Real Estate Agent
Date	Date Sold
Distance	Distance from CBD in kilometers
Postcode	Postcode
Bedroom2	Scraped number of bedrooms (from different source)
Bathroom	Number of bathrooms
Car	Number of carspots
Landsize	Land size in metres
BuildingArea	Building size in metres
YearBuilt	Year the house was built
CouncilArea	Governing council for the area
Lattitude	Lattitude
Longitude	Longitude
Regionname	General Region (West, North West, North, North east ...etc)
Propertycount	Number of properties that exist in the suburb

A histogram of the data is shown in Figure 1. The skewness is 2.59 and Kurtosis is 13.08 A positive value of skewness means that the distribution is positively right-skewed i.e, has a long right tail.

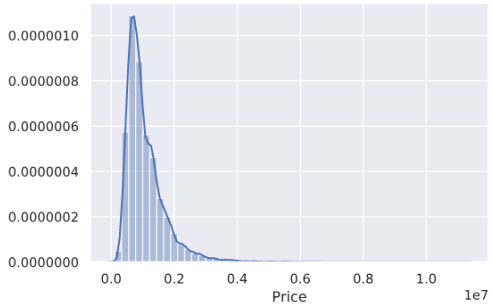

Fig. 1. Positively skewed data

3 Methodology

3.1 Data Cleaning and Preprocessing

The data in the raw form is not suitable for predictive modeling. The features in the dataset are in various formats. It has numerical data such as number of rooms, distance, number of bathrooms, number of cars that can be parked, land size, building area etc., as well as categorical features such as house type, method sold, seller, council area etc. Numerical features are compatible with regression models, but to make categorical features compatible, we will have to convert them to suitable formats. The samples seem to have missing data which needs to be addressed. We also need to check the dataset for outliers. All these steps are explained in detail in the following sections.

3.2 Incomplete, Missing Data

A dataset can have missing values due to corruption in the way it was collected. It is important to handle these missing values to achieve successfull results. We analyzed our dataset and there appears to be missing information in the dataset. There is significant amount of missing values in Price (21.83%), Bedroom2 (23.57%), Bathroom (23.59%), Car (25.03%), Landsize (33.8%), BuildingArea (60.57%), YearBuilt (55.38%), CouncilArea (0.008%), Lattitude and Longitude (22.8%), Regionname (0.008%) and Propertycount (0.008%).

Case 1 Handling Price: Price is the target feature in our problem. So, we remove any rows which have price as null.

Case 2 Handling Lattitude and Longitude: We used Geopy [2] to handle the missing values for lattitude and longitude. Geopy is a python client for several popular geocoding web services. It makes it easy for python developers to locate the coordinates of addresses across the globe. For this dataset, we combined two features- Suburb and Address, to form the complete address of the property. Then this address was passed to geolocator to obtain the spatial coordinates. However, there were still around 500 values missing from these two columns which the API was not able to fetch, so we fill those missing values with the median.

Fig. 2. Geographical view of the properties in Melbourne

Case 3 Handling Postcode: There was only one value missing for postcode. We use the Geopy API, used for handling missing values for lattitude and longitude, to obtain the Postcode for this sample and fill it with the correct value.

Case 4 Handling Distance: There was only one value missing for distance, since it is a numerical value, we can replace it with the mean value of distance in the dataset.

Case 5 Handling CouncilArea, Regionname and Propertycount: These three features are categorical and only three rows have missing values for these features. So we replace it with the mode value of the column in the dataset.

Case 6 Handling Bathroom, Car, Landsize, BuildingArea: The values for these features are filled with mean values by imputation. The imputed values might not be accurate for some samples, but it usually gives more accurate models than dropping the columns completely.

Case 7 Handling YearBuilt: YearBuilt is a numeric data but we cannot have decimal values for year, so instead of imputing the missing values with mean, we impute using the median.

After handling of missing values, we are left with 27,247 samples and 21 features in our dataset.

3.3 Duplicate Features

According to the dataset description, both ‘rooms’ and ‘Bedroom2’ represent the number of rooms in a house. However, the values for ‘Bedroom2’ has been collected from a different source and is less likely to be reliable. The difference in the values of these two columns were found to be minimal, so ‘Bedroom2’ feature is eliminated from our analysis.

Similarly, The columns ‘Suburb’ and ‘Address’ are nothing but Nominal representation of lattitude and longitude, so it is safe to drop the Suburb and Address columns from consideration.

3.4 Outlier Detection and Removal

Most machine learning algorithms are sensitive to the distribution of feature values in the dataset. Outliers in the dataset can skew the training of models and could lead to less accurate models. They could also represent incorrect data or samples that are irrelevant to the problem.

Building Area: We did a statistical study about the data and observed that the minimum value for building area is zero, which seems incorrect. We found out that a total of 61 samples had zero building area. These samples are invalid and can be removed from the dataset.

Also, the maximum value for building area is 44,515 (see fig.1). If we observe this sample, we notice that the landsize for the same is 44,500 which is less than the building area. The building area cannot be greater than landsize, and keeping in mind that the sample has 5 bedrooms, 3 bathrooms and 5 car spots, it is fair to assume that the sample could be incorrect so we can drop it.

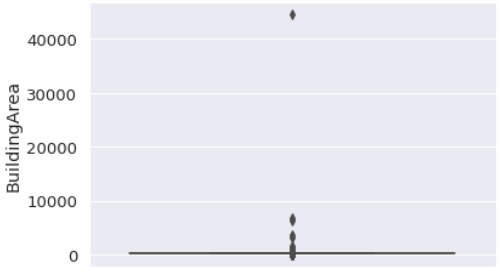

Fig. 3. Boxplot of the feature Building Area

Landsize: We observed that there were a few samples that had landsize zero. These samples turned out to be invalid so we can drop these samples.

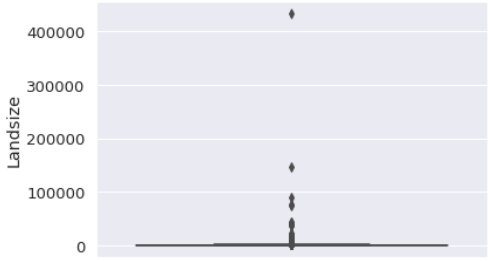

Fig. 4. Boxplot of the feature land size

Bathroom: When we get the count for unique values of the feature, we note that there are few samples that have as many as 9 bathrooms. If we have a close look at the samples which have greater than 7 bathrooms, we are able to narrow down the dataset to 4 rows. The number of bedrooms in the houses which had a lot of bathrooms were also around 8-16, so the large number of bathrooms seems fair. Also, the price for such properties are also around $2,200,000 - $5,000,000 which justifies that the samples are valid.

YearBuilt: For this feature, we want to make sure that there is no house with the value which is greater than 2019, which is the current year. We found a sample with the year built value 2106 which seemed like a typographical error. So we replace the value 2106 with 2016.

3.5 Feature Engineering

Age: We can caluclate the age of the property, i.e, the number of years since the building was built from the feature YearBuilt. We can then use age as a feature rather than using YearBuilt itself. This feature indicates how old or new the property is, and hence it is useful in estimating the price.

Year and Season: The Date feature in the dataset indicates the time of the year when the property was sold. This helps us understand what the market prices are depending on the season. We extract two features – Year and Season from the original feature Date. Getting the year from the date is straightforward. We get the season by first calculating the day of the year from the date and then specifying a range (70-150 for autumn, 150-240 for winter, 240-330 for spring and everything else for summer).

Categorical Data: Machine learning models are based on equations and are not good with categorical data. We need to convert categorical data into numerical

data so that we can include those features in our regression models. One way to do so is assigning numbers to the text values. However since 1>0 and 2>1 (and so on), the equations in the model might believe that the text which was assigned as 1 has higher value that the text which was assigned as 0. But this is not the case, these text values are just categories and there is no relational order between them. So, we use dummy variables to represent the caregorical features in the data. Dummy variables take the value 0 or 1 to indicate the absence and presence of the feature. Instead of having 1 column for the features, we have as many columns as the number of categories. We can use python library called pandas which provides the get_dummies method to achieve the same. The features that we need to transform are- Type, Regionname and season.

3.6 Feature Extraction

Correlation Analysis: It is a measure that helps us analyze how two features are linearly related to each other. We want to evaluate which features have the highest impact towards our target prices. To do this, we plot the heat map (table123) which describes the correlation coefficients of our variables. The map shows high dependency of price with features such as rooms, bathroom, car, building area and age. The variables that are highly correlated with the target are likely significant in regression analysis.

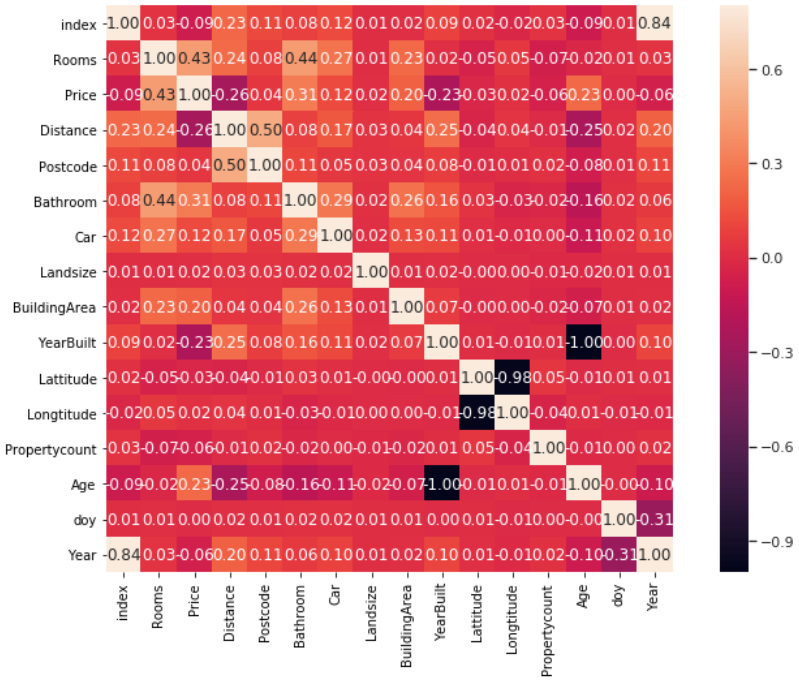

Fig. 5. Correlation Analysis

SelectKBest: The classes in the sklearn.feature_selection module can be used for feature selection to improve accuracy, reduce overfitting and reducing training time. It selects features according to the k highest scores. The function takes two arrays X (features) and y (target) and returns scores [3]. The method get_support returns the features selected. This algorithm shows us that we should select Distance, Land size, building area, age, property count, house type and region area.

So, we have narrowed down our features using correlation analysis and SelectKBest. But none of these features give an idea about the location of the property or the trend of market depending on the season, so we include four more features- Longitude, Lattitude, Season and year to make sure that we are not losing any information.

3.7 Useless features

In the previous section we have selected the features that can be used for regression modelling. Now we can drop the features that are not usefull from consideration.

We have the features Longitude and Lattitude which are nothing but spacial coordinates derived from combining suburb and address, so it is safe to drop the features suburb and address. We have fetched the feature age from the year built, so we can drop yearbuilt. We have computed the columns year and season from the date feature, so we can drop the date.

3.8 Final features

The final features to be used in all the regression models are -Rooms, Distance, Bathroom, Car, Landsize, BuildingArea, Lattitude, Longtitude, Propertycount, Age, Year, h, t, u, Eastern Metropolitan, Eastern Victoria, Northern Metropolitan, Northern Victoria, South-Eastern Metropolitan, Southern Metropolitan, Western Metropolitan, Western Victoria, autumn, spring, summer, winter

3.9 Normalization

At this point, all of our data has been converted to numeric form. But they are not on the same scale. When we are working with a regression problem, it is important to scale the features to the same range, which will also make their variance in the same range. This is doneso that the order of magnitude of the features are same. To achieve this we use the Min-Max Scaling technique. In this, the data is scaled

to a fixed range- usually 0 to 1 to achieve Gaussian with zero meand and unit variance. We have used sklearn.preprocessing.MinMaxScaler [4] to achieve this.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}(new_{max} - new_{min}) + new_{min}$$

4 The Models

4.1 Linear Regression

Linear regression is a simple and earliest predictive method in supervised learning. It uses a linear combination of independent features to estimate a continuous output, the dependent variable. Linear Regression fits a linear model with coefficients $w = (w_{1,\ldots,} w_p)$. It aims at calculating the coefficients in a way that the mean squared error is minimized and the responses are predicted by linear approximation [5]. The disadvantage of linear regression is that its performance is influenced by the presence of outliers and random errors, producing a large variance.

4.2 Decision Tree Regressor

A Decision Tree Regressor works on the same principal as that of Decision Tree Classifier with the difference that the target feature values can now take continuous values [6]. It searches for the feature that splits the target feature most appropriately. The dataset is then divided along the values of this feature and the process is repeated until some stopping criteria. The average target feature values is returned as prediction at the leaf node.

4.3 Nearest Neighbors

The principle behind nearest neighbor method is to find a predefined number of training samples closest in distance to the new point, and predict the label from these [7]. The number of samples (k) is an integer value defined by the user. There are several ways of measuring distances between points a and b in d dimensions- with closer distances implying similarity- such as Minkowski Distance Metric, Weighted Minkowski, Cosine Similarity, Kullback Leibler Divergence etc. In our nearest neighbors regression model, we have taken the value of k as 5.

4.4 Bagging Meta-Estimator

Ensemble learning is a machine learning technique that combines many different models into one predictive model. It revolves around the idea of voting and averaging.

In Bagging meta-estimator, subsets of the original training set are picked and then

their individual predictions are aggregated to form a final prediction. This reduces the variance of a base estimator (Decision Tree Regressor in our case) by introducing randomization into its construction and then making an ensemble out of it. [8].

4.5 Random forest Regressor

In random forests, each tree in the ensemble is built from a sample drawn with replacement from the training set. The split that is picked is the best split among a random subset of the features. Due to this randomness, the bias of the forest usually slightly increases and due to averaging, the variance decreases [9].

5 Experiments

5.1 Training and Test datasets

The dataset split into training data and test data. The training set contains a known output and is used for learning. The test dataset is used to test the model's prediction. We do this using the method train_test_split provided by the Scikit-learn library. We split the data set in 80:20 ratio, 80% for the training and 20% for testing.

5.2 Cross Validation

It is a method that estimates test error by holding out a subset of training data from the fitting process. K-Fold Cross Validation is where a given data set is split into K number of sections/folds and then each fold is used as a testing set at some point. In our experiments, we have taken the value of k as 10.

5.3 Evaluation Metrics

Mean Squared Error (MSE) is the mean of the squared errors.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2$$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors (MSE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2}$$

Mean Absolute Error (MAE) is the average of the absolute difference between the predicted values and observed values. It is a linear score which means that all the differences are weighed equally in average.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'|$$

R squared (R^2) explains how well the selected independent variable explain the variability in the dependent variable. Higher the MSE, smaller the R squared value and poorer the model.

$$R\ Squared = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{\sum_{i=1}^{n}(y_i - Y_i)^2} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - Y_i)^2}$$

6 Results

6.1 Train-Test split Results

The results from the various regression models, mentioned in the previous sections, on the Melbourne Housing dataset is shown in table 2 and table 3.

Table 2. Train Test split regression results of various models

Model	MSE	RMSE	MAE	R^2
Linear Regression	0.00148	0.038	0.025	0.56
Decision Tree	0.00120	0.034	0.020	0.63
Nearest Neighbors	0.00129	0.035	0.021	0.60
Bagging Regressor	0.00059	0.024	0.014	0.81
Random Forest Regressor	0.00061	0.024	0.014	0.81

Table 3. Train Test split regression results of various models

Model	Pearson Correlation coefficient	Spearmans correlation coefficient
Linear Regression	0.75	0.81
Decision Tree	0.81	0.87
Nearest Neighbors	0.78	0.82
Bagging Regressor	0.90	0.91
Random Forest Regressor	0.90	0.91

6.1 Cross Validation Results

The results from the k-fold cross validation on various regression models, mentioned in the previous sections, on the Melbourne Housing dataset is shown in table 4.

Table 4. K-Fold cross validation Results

Fold	Linear Regression	Decision Tree	Nearest Neighbors	Bagging Regressor	Random Forest Regressor
1	0.0274	0.0279	0.0302	0.0213	0.0216
2	0.0275	0.0269	0.0284	0.0205	0.0206
3	0.0259	0.0247	0.0268	0.0183	0.0183
4	0.0251	0.0215	0.0232	0.0163	0.0164
5	0.0246	0.0207	0.0207	0.0150	0.0150
6	0.0237	0.0187	0.0195	0.0146	0.0146
7	0.0238	0.0183	0.0192	0.0142	0.0142
8	0.0259	0.0214	0.0212	0.0164	0.0164
9	0.0250	0.0185	0.0205	0.0143	0.0144
10	0.0255	0.0205	0.0213	0.0145	0.0145

The statistical tests are conducted using Friedman's test. The default assumption, or null hypothesis, is that the multiple paired samples have the same distribution. A rejection of the null hypothesis indicates that one or more of the paired samples has a different distribution.

Fail to Reject H_0: Paired sample distributions are equal.

Reject H_0: Paired sample distributions are not equal.

Overall Comparison (shown in table 4) gives the statistics value as 35.44 and p

value of 0.000. The value of apha is 0.05. The distributions are different. Reject H_0.

Comparison of Linear Regression, Decision Tree Regressor and Nearest Neighbors gives the statistics value as 8.600 and p value of 0.014. The value of apha is 0.05. The distributions are different. Reject H_0.

Comparison of Decision Tree Regressor, Bagging Regressor and Nearest Neighbors gives the statistics value as 16.800 and p value of 0.000. The value of apha is 0.05. The distributions are different. Reject H_0.

Comparison of Random Forest Regressor, Bagging Regressor and Nearest Neighbors gives the statistics value as 20.00 and p value of 0.000. The value of apha is 0.05. The distributions are different. Reject H_0.

Comparison of Random Forest Regressor, Bagging Regressor and Linear Regression gives the statistics value as 20.00 and p value of 0.000. The value of apha is 0.05. The distributions are different. Reject H_0.

Comparison of Random Forest Regressor, Decision Tree Regressor and Linear Regression gives the statistics value as 18.200 and p value of 0.000. The value of apha is 0.05. The distributions are different. Reject H_0.

Fig. 6. Train and Test error accross folds for Linear Regression

Fig. 7. Train and Test error accross folds for Decision Tree Regressor

Fig. 8. Train and Test error accross folds for 5 Nearest Neighbour

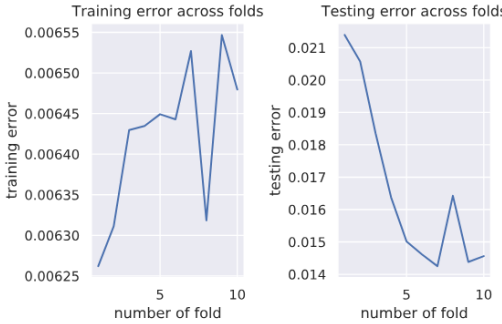

Fig. 9. Train and Test error accross folds for 5 Bagging Regressor

Fig. 9. Train and Test error accross folds for 5 Random Forest Regressor

7 Discussion

The data in the raw format required a lot of pre processing. It contained a lot of missing values that needed to be handled. Some could be derived from other features, for example the spatial coordinates could be derived from suburb and address. But there was no way to find the exact values for other features hence their values had to be imputed. There were some outliers in the dataset which were easily identified and removed.

It is observed that there is comparable difference in the performances of the models. The most accurate model turned out to be that of Ensembles. We also note that the performance of the Random Forest Regressor and Bagging Regressor is almost the same.

The prices of the house are impacted by various other factor such as if there are schools in the community, if the society is pet friendly, if the society is child safe, number of thefts happened in the past etc. These are some factors which were not included in the dataset and actually impact the price of a property in real life. However, the most important factors such as number of rooms, number of bathrooms, area, number of carspots etc are covered in the dataset.

8 Conclusion

With the help of various preprocessing techniques, such as Handling mising values, Outlier Detection, Feature extraction, Feature selection etc., we cleaned the raw Melbourne Housing data provided on kaggle website. These techniques helped in the analysis of the dataset and identify which feature of the house were most likely to affect the market price of the house. Using various regression

models such as linear regression, decision trees, nearest neighbors, bagging regressor and random forest regressor, we were able to predict the value of the house price given certain features. Furthermore, we were able to improve the model's accuracy using K-Fold cross validation techniqe. Another aim of this project was to examine various regression models and compare their performance given that the same features were fed as an input for predictions. This was achieved using Friedman's test. Overall, there was significant difference in the results of the models.

9 Future Work

Th next step for this project could be tuning the hypermeters of the models (such as number of estimators, depth of the trees etc) to improve the accuracy. Techniques such as Grid search CV provided by sklearn.model_selection could be used for this. We could also work on refining the features in the dataset and include some real life factors such as number of schools in the community, if the society is pet friendly, if the society is child safe, number of thefts happened in the past etc

References

1. Domain, https://www.domain.com.au/
2. GeoPy's documentation, https://geopy.readthedocs.io/en/stable/#nominatim
3. Sklearn documentation, https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
4. Sklearn documentation, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
5. Sklearn Linear Models, https://scikit-learn.org/stable/modules/linear_model.html
6. Sklearn Decision Tree Regressor, https://scikit-learn.org/stable/modules/tree.html#regression
7. sklearn Nearest Neighbors, https://scikit-learn.org/stable/modules/neighbors.html
8. sk learn Ensemble, https://scikit-learn.org/stable/modules/ensemble.html#bagging-meta-estimator
9. sklearn Random Forests, https://scikit-learn.org/stable/modules/ensemble.html#random-forests