House Price Prediction: Advanced regression techniques

Manel ALOUI

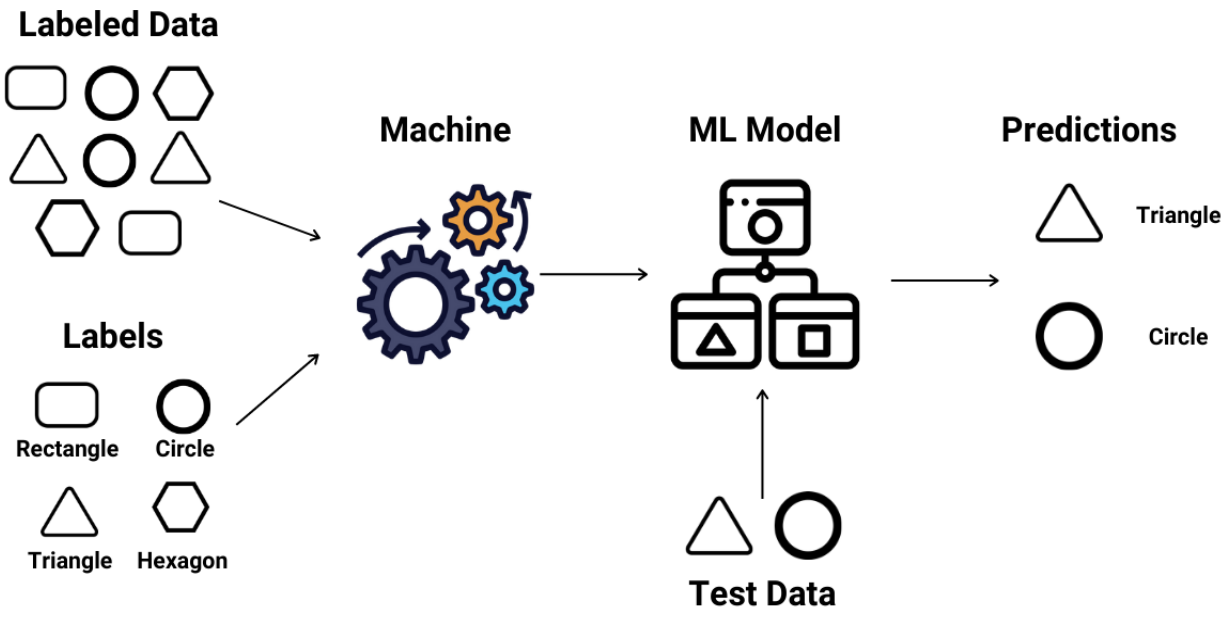
10/10/2022

Plan

- Introduction
- Project Specification
- Pipeline
- Conclusion

Introduction

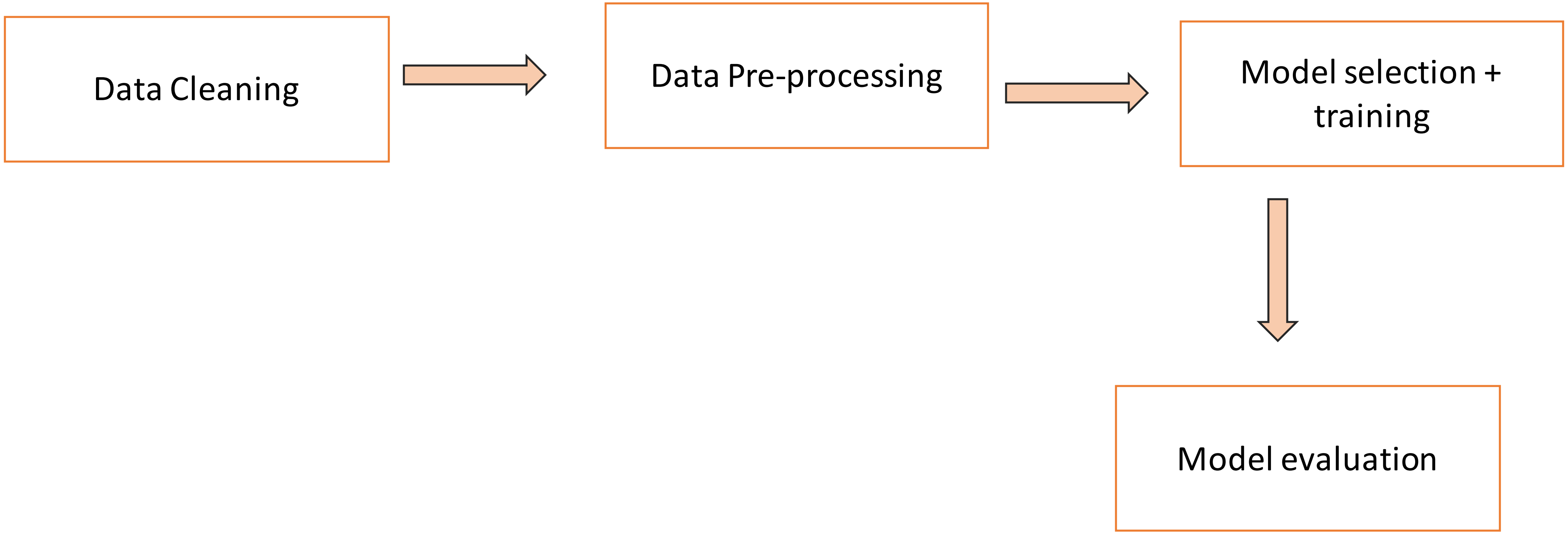
- Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit this data into models that can be understood and utilized by people later.
- Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data to allow it to find structure within its input data.
- In this project we will focus on **supervised Learning**



Project specification

- The goal of this project is to predict the house price based on several variables.
- The dataset used is open source through kaggle https://www.kaggle.com/c/house-prices-advanced-regression-techniques .
- There are 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa with 1460 observations in the training set.

Pipeline



Data Cleaning

Data Pre-processing

Model selection + training

Model evaluation

Data cleaning

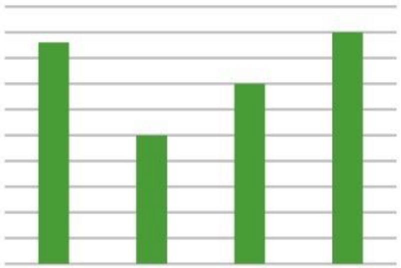
Aim: The main goal of Data Cleaning is to identify and remove errors & duplicate data, to create a reliable dataset. This improves the quality of the training data for analytics and enables accurate model results .

Steps:

- To get more insights we decided to merge train and test data together but without doing shuffle on the index (so that we can re-verse back to get the same test data).
- Checking for missing values: there are two types of values: categorical and continuous, and dealing with their missing values is different.

Continuous

Age, height, distance, temperature…

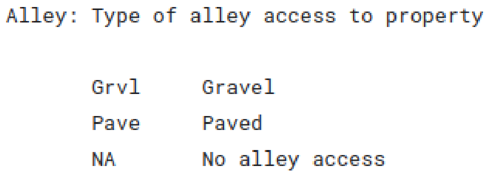


Categorical
(also called "discrete")

Age group, sex, number of siblings, citizenship, race…

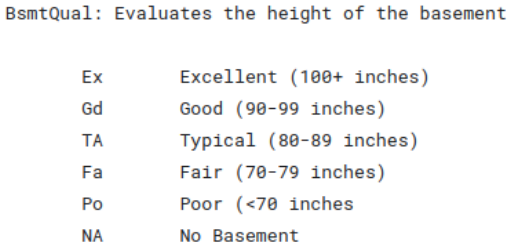
1. Fill Categorical missing values: When checking the description of the data, there are some columns having "NA" but this do mean a specific values so we shouldn't treat it as a missing values.

Example:



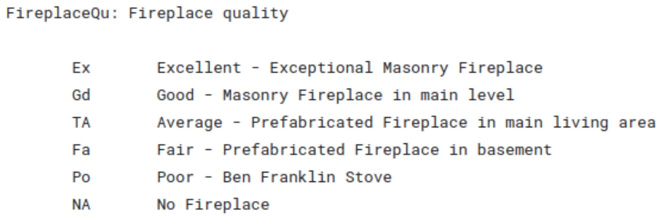
Alley: Type of alley access to property

Grvl	Gravel
Pave	Paved
NA	No alley access



BsmtQual: Evaluates the height of the basement

Ex	Excellent (100+ inches)
Gd	Good (90-99 inches)
TA	Typical (80-89 inches)
Fa	Fair (70-79 inches)
Po	Poor (<70 inches
NA	No Basement



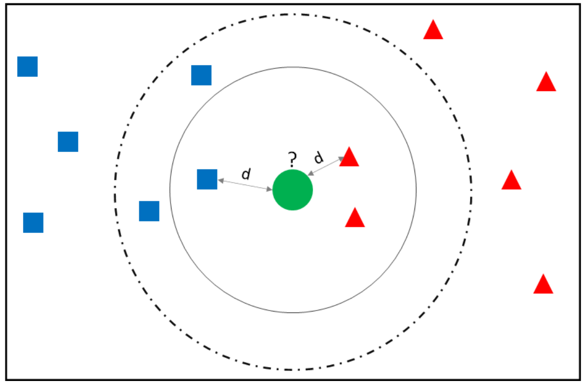
FireplaceQu: Fireplace quality

Ex	Excellent - Exceptional Masonry Fireplace
Gd	Good - Masonry Fireplace in main level
TA	Average - Prefabricated Fireplace in main living area
Fa	Fair - Prefabricated Fireplace in basement
Po	Poor - Ben Franklin Stove
NA	No Fireplace

If a column has "NA" as a specific meaning, we impute missing values with a constant.
Else we impute with the mode (The most frequent number).

2. Fill Numerical missing values: We impute with numerical missing values with the KNN Imputation technique:

This technique utilizes the k-Nearest Neighbors method to replace the missing values in the datasets with the mean value from the parameter 'n_neighbors' nearest neighbors found in the training set.
By default, it uses a Euclidean distance metric to impute the missing values.



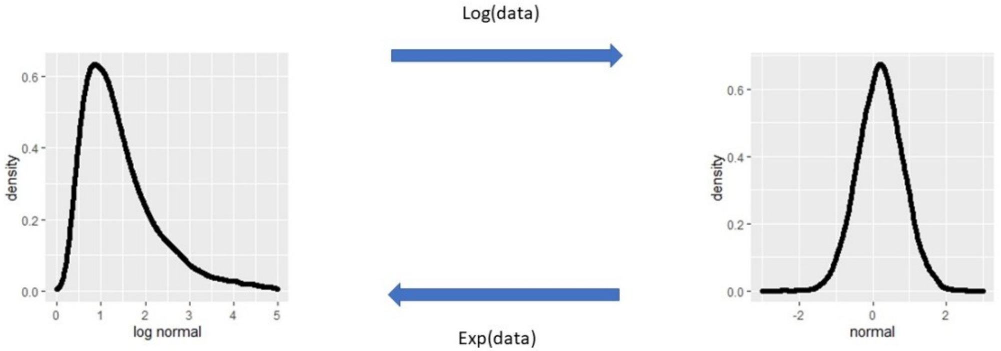
Data Pre-Processing

- Feature transformation
- Encoding categorical data
- Feature scaling

1/ Feature transformation

- We start by checking the skewness of the data
- Skew is the degree of distortion from a normal distribution, if we have skewed data, the tail region may act as an outlier for the statistical model, and we know that outliers adversely affect a model's performance, especially regression-based models
- For skewed features we apply log transform which can help

  to fit a very skewed distribution into a Gaussian one
- Fo cyclical features like Cosine Transform for cyclical Features
- For the target we apply the log transformation .



Log normal distribution

Normal distribution

2/ Encoding categorical data:

Since most machine learning models only accept numerical variables, preprocessing the categorical variables becomes a necessary step. We need to convert these categorical variables to numbers such that the model can understand and extract valuable information.

We do this with get_dummies function from Pandas library

3/ Feature scaling:

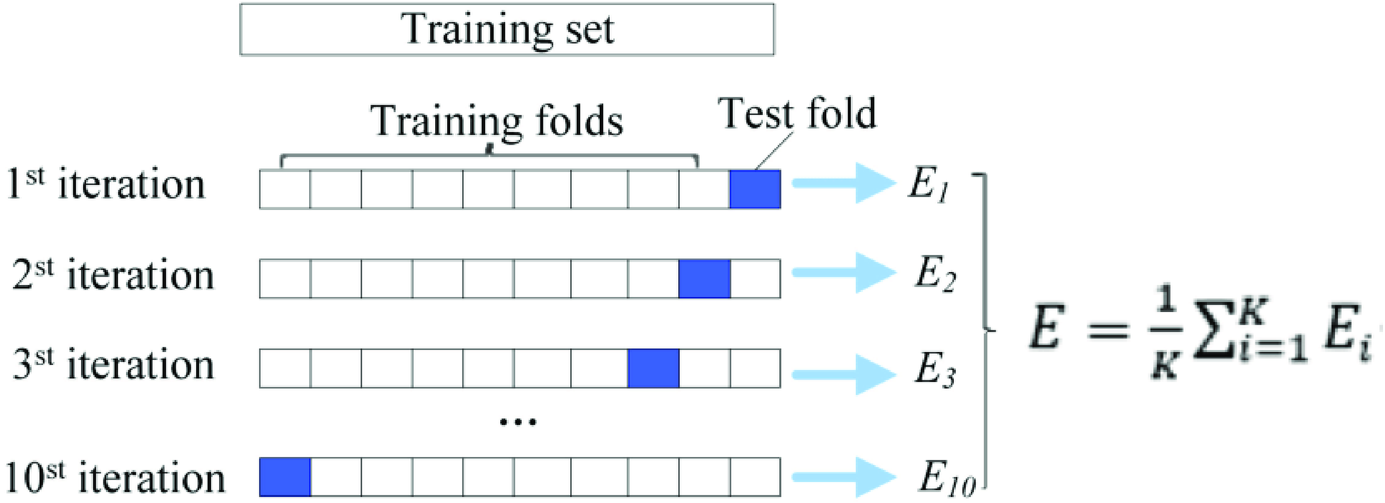
Feature scaling is the process of normalizing the range of features in a dataset. Real-world datasets often contain features that are varying in degrees of magnitude, range and units. Therefore, for machine learning models to interpret these features on the same scale, we need to perform feature scaling.

We use Standardscaler for this

--> With all these steps done, we have our dataset ready to be trained and fed to the model.

Model selection + Training

- Having the pre-processed data in hand we split again the data into its first indexed test train set and use the trainset to train the model.
- During the training we apply the technique of cross validation to detect overfitting, ie, failing to generalize a pattern.



- For model selection, Pycaret helps us to try and select a set of models.
- We apply it and try with top 5 models.
- To compare well the model's result, it's recommended to start with a Baseline model: in our case it's: CatBoostRegressor()
- Then we train 4 other models and combine predictions together: Bagging Ensemble.
    - CatBoostRegressor(verbose = 0)
    - BayesianRidge()
    - OrthogonalMatchingPursuit()
    - Ridge()
    - LGBMRegressor()

Model Evaluation

- A big step that shouldn't be skipped: We must apply the exponentials to the results prediction in order to omit the log .
- To evaluate our model, we use the Root Mean Square Error .
- The RMSE estimates the deviation of the actual y-values from the regression line.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Model	RMSE
Baseline Model	0.12595
Bagging Ensemble Model	0.12395

Conclusion

- In this project we predict the price of a house given a set of features, the dataset was given by Kaggle
- This work could be improved by:
  - Applying some feature Engineering: creating new features from existing one so that the model could understand better the relationship between independent value and dependent values
  - Do some Hyper parameter tuning with grid Search.
  - Try with some deep learning models.