Domain Background

since I am still have a low knowledge and experiences with Machine Learning field, because they have not in the field for too long, I hope to proposal my project in a field which is expanding day by day. So I picked a project related to real estate from kaggle, a helpful area for Machine Learning Engineer and related fields.
Project: House Prices: Advanced Regression Techniques.
link on kaggle:
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview
This project with around 80 features in it's dataset, so through this exercise, I would be able to have a good knowledge of what impact the prices of house using features in dataset.
I would also note that the Ames Housing dataset was complied by Dean De Cock for use in data science education, visit the link below:
http://jse.amstat.org/v19n3/decock.pdf

Problem Statement

Based on the features of the dataset I have a goal to predict a house price using what I am learning from available features,This is a supervised regression task. Kaggle suggested Random Forest and Gradient Boosting as ways to doing this task. However, I think the task is more comprehensive than plugging train and test datasets into sklearn. There will be data exploration, PCA, grid search, and MLP components. Some of different ways we can work with this dataset listed below:
Quantifiable: different ML techniques can be used to express how to find housing prices in math or logical terms to be more understandable.

Measurable: using root mean square error, for measuring our model effectiveness.

Datasets and Inputs

this dataset for this exercise generated from the Ames City Assessor's Office. Variables that required special domain knowledge and variables that were present for weighing and adjustment purposes were removed. Data attribute contains 79 features (23 nominal, 23 ordinal, 14 discrete and 20 continuous), 1 target, and 1459 observations, because it is sensitive information, such as address and others like zip code, were never present, it will be difficult to improve model performance, by introducing external data and joining with the existing dataset. You can refer to attached features in dataset file.

Our goal of this exercise to predict sale price, which is the target of our dataset, It is a continuous variable and describe the final sales price of the property. So I will working on exploring all of the features, 23 nominal, 23 ordinal, 14 discrete and 20 continuous, and their potentials to predict Sale Price.

Because dataset fro this exercise came from kaggle, I will be using the given training dataset as my train, validation, and test datasets. I will split the data into train and test (80:20, random split), then split the remaining data further into train and validation (80:20) for my model optimization process.

Solution Statement

The solution will be house price predictions, which is continuous. Because of the selected ML, I will be able to quantify the solution in math or logical terms. I will be able to evaluate our predictions with the validation and test datasets to calculate the error measurement (root mean square error) and replicate the solution because the optimal model will be preserved. Keep in mind, this is an exercise to minimize the error term on the prediction/solution. Here's a list of algorithms and techniques.

I want to do this:

Data exploration, including feature engineering and PCA:

- Visualize distributions for each feature.
- Find outliers and determine if they need to be removed.

Supervised model selections leveraging grid search/k-folds including:

- Decision Tree Regressor
- Random Forest Regressor
- MLP Regresso
- KerasRegressor and others

Benchmark Model

I will use the results from Decision Tree Regressor as a benchmark. All the model performance will be compared to the Decision Tree Regressor, given they are fitted to the same train and validation set.

Evaluation Metrics

The main evaluation metric will be root mean square error:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Root mean square error is a standard evaluation for modeling exercises with target that is continuous. I will split the provided dataset to do train, validation and test sets. I will choose the modeling hyperparameters based on how the root mean squared error performed on train and validation datasets. After optimizing the hyperparameters, I will then choose the models based on test data performance.

Project Design

- Explore data – I will take a look at how the data is distributed by looking at the max, min, median and, and histogram.
- Cleaning data – I will look at how each feature is presented, and how it relates to out target. I will also look at how the outliers and missings data should be managed.
- Prepare data – I will use sklearn.model_selection.train_test_split to randomly split the data into training and test sets.
- Feature treatments – with 79 features, perhaps PCA and feature engineering/transformation are required. I'll take a look at how these methodologies could help reduce the dimensionality of this exercise. Note that different model may require different feature treatments.
- Model selection – given this is a regression exercise, I will experiment with different modeling algorithms. To optimize each model's hyperparameters, I will use grid search in choosing hyper parameters. During each optimization exercise, I will look at how the error terms improve as well as if the models are overfitting. I will use visualization to illustrate how I reach various decisions along the exercise.

  Some of the models I will use:
    - DecisionTreeRegressor
    - RandomForestRegressor
    - KNeighborsRegressor
    - GradientBoostingRegressor
    - MLPRegressor