Milestone 3 - Project Report

[House Prices - Advanced Regression Techniques]

(Submitted by - Lakshya Rathi - E23CSEU0034)

Domain: The project is focused on the Kaggle house prices advanced regression competition. The goal is to predict the sale price of houses based on various features provided in the dataset.

Abstract

The increasing and decreasing occurrence of house prices changes from time to time. There are more reasons that effect the fluctuation of house prices. Some are build year, location, physical amenities, size of house etc. Predicting the value of house (Sale price) helps the customers to take right choice of buying the house. Machine learning is being adapted for various fields that could build prediction model and estimate the outcomes. In this paper, we are contemplating the issue of rise and fall of house rates as a regression problem. Regression is a process that aims to predict the correlation between target dependent feature and a sequence of other changing independent features.

Introduction

Day by day, the tendency of people liking to improve their living standards has been increasing. This is heading more of demand for houses. But, the problem is that customers may not know how much worth exactly a house to purchase. It may leads to take wrong decisions. Predicting the price or rate of a house is called house sale price prediction. It helps buyers and sellers. The price of the house varies now and then. The reason behind the changing value of house is based on many features. Some features are location, size of house, bedroom units, and count of storeys, living area, bathrooms, garage size, house age, type of roof and other utilities. These are considered as independent features in which no feature has relation with other feature. The target feature to predict is Sale price. This is considered as dependent feature in which its value effects in changing the values of independent features. Machine learning allows the machines to learn and to perform operations by themselves instead of inputting instructions explicitly. The machine learning project lifecycle workflows is as shown in below figure.

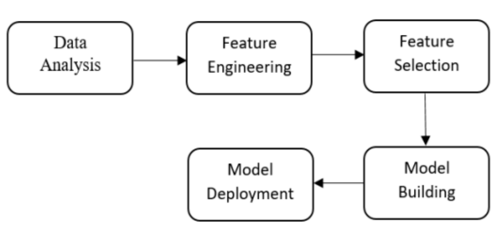


Figure 1: Machine learning project life-cycle

The first phase is Data Analysis. It is the process of visualizing data in means of graphs, finding missing values and observing correlations between features. Second phase is Feature Engineering. It is an activity of converting raw datasets into features which helps in improving the performance of machine learning techniques. Next one is feature selection. It is defined as a way of picking the most important features that effect more to predict the resultant outcome. Model building is defined as using machine learning techniques that enables model to learn from data without giving instructions. Model Deployment is the process of integrating built ML model to dynamic environment to make predictions from historical data. Prediction in machine learning can be defined as generating an output from dataset input that is applied to a model. The model that best fits to a dataset implies an accurate prediction. We observed and implemented this problem using Regression analysis. Regression is a machine learning technique in which it can be performed whenever we need to model numerous independent features and to predict continuous dependent feature. Since predicting house price that is based on many independent changing features.

2. Related Work

Recent studies in house price prediction demonstrate diverse methodological approaches, with each researcher bringing unique insights. Ankita's project stands out with its comprehensive exploration of nine models, ultimately achieving an RMSE of 0.3769 using a Stacked Regressor that combined Random Forest, SVR, KNN, and Ridge models. Her preprocessing included handling missing values, outlier removal, and log transformation of the target variable. Joanna Broniarek focused on boosting techniques and ensembles, emphasizing extensive feature engineering and normalization, while Tracy Renee specialized in XGBoost optimization for Kaggle competition improvements.

The field shows significant diversity in approaches: Sifei Lu developed a hybrid Lasso-Gradient Boosting technique, while Ayush Varma explored neural networks for prediction accuracy. Adyan Nur introduced Particle Swarm Optimization alongside regression analysis, and G. Naga Satish focused on enhancing Lasso Regression functionality. Nehal N Ghosalkar approached the problem through Linear Regression with a customer-centric focus, while D. Banerjee innovated with classification-based techniques for price direction prediction.

More advanced approaches came from Bruno Klausde, who combined Recurrent Neural Networks with Random Forest ensembles, and T.D. Phan, who integrated SVM with neural networks for market trend prediction. Rakesh D.'s WARSE paper provided valuable insights into model comparisons, while Sayan Putatunda applied Random Forest and Gradient Boosting specifically to the Indian real estate market.

Across all projects, researchers consistently utilized Python-based tools (sklearn, pandas, numpy, seaborn, matplotlib) and emphasized thorough preprocessing steps. Common elements included feature engineering, handling missing values, normalizing data, and treating categorical variables. This collective research demonstrates the evolution from basic regression techniques to sophisticated ensemble and hybrid approaches in house price prediction.

3. Dataset

House Price Dataset For experimental purpose, we imported House price datasets from https://www.kaggle.com. Dataset sample is as shown
[House Prices - Advanced Regression Techniques | Kaggle]

The dataset consists of a training set with around 1,460 records and 81 columns/features. The test set does not include the sale price column. The key features include numerical values like lot frontage, as well as categorical features like MS Zoning.

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl

The description of dataset is representing in below

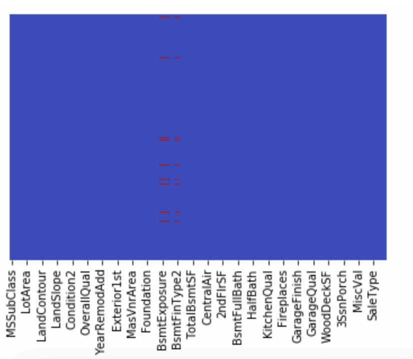
Variable	Data type	Data description
Id	numeric	Identity number
MSSubClass	numeric	The building class
MSZoning	text	The general zoning classification
LotFrontage	numeric	Linear feet of street connected to property
LotArea	numeric	Lot size in square feet
Street	text	Type of road access
Alley	text	Type of alley access
LotShape	text	General shape of property
LandContour	text	Flatness of the property
Utilities	text	Type of utilities available
LotConfig	text	Lot configuration
LandSlope	text	Slope of property
Neighborhood	text	Physical locations within Ames city limits
Condition1	text	Proximity to main road or railroad
Condition2	text	Proximity to main road or railroad (if a second is present)
BldgType	text	Type of dwelling
HouseStyle	text	Style of dwelling
OverallQual	numeric	Overall material and finish quality
OverallCond	numeric	Overall condition rating
YearBuilt	numeric	Original construction date
YearRemodAdd	numeric	Remodel date
RoofStyle	text	Type of roof
RoofMatl	text	Roof material
Exterior1st	text	Exterior covering on house
Exterior2nd	text	Exterior covering on house (if more than one material)
MasVnrType	text	Masonry veneer type
MasVnrArea	numeric	Masonry veneer area in square feet
ExterQual	text	Exterior material quality
ExterCond	text	Present condition of the material on the exterior
Foundation	text	Type of foundation
BsmtQual	text	Height of the basement
BsmtCond	text	General condition of the basement
BsmtExposure	text	Walkout or garden level basement walls
BsmtFinType1	text	Quality of basement finished area
BsmtFinSF1	numeric	Type 1 finished square feet
BsmtFinType2	text	Quality of second finished area (if present)
BsmtFinSF2	numeric	Type 2 finished square feet
BsmtUnfSF	numeric	Unfinished square feet of basement area
TotalBsmtSF	numeric	Total square feet of basement area
Heating	text	Type of heating
HeatingQC	text	Heating quality and condition
CentralAir	text	Central air conditioning
Electrical	text	Electrical system
1stFlrSF	numeric	First Floor square feet
2ndFlrSF	numeric	Second floor square feet
LowQualFinSF	numeric	Low quality finished square feet (all floors)
GrLivArea	numeric	Above grade (ground) living area square feet
BsmtFullBath	numeric	Basement full bathrooms
BsmtHalfBath	numeric	Basement half bathrooms
FullBath	numeric	Full bathrooms above grade
HalfBath	numeric	Half baths above grade
Bedroom	numeric	Number of bedrooms above basement level
Kitchen	numeric	Number of kitchens
KitchenQual	text	Kitchen quality
TotRmsAbvGrd	numeric	Total rooms above grade (does not include bathrooms)
Functional	text	Home functionality rating
Fireplaces	numeric	Number of fireplaces
FireplaceQu	text	Fireplace quality
GarageType	text	Garage location
GarageYrBlt	numeric	Year garage was built
GarageFinish	text	Interior finish of the garage
GarageCars	numeric	Size of garage in car capacity
GarageArea	numeric	Size of garage in square feet
GarageQual	text	Garage quality
GarageCond	text	Garage condition
PavedDrive	text	Paved driveway
WoodDeckSF	numeric	Wood deck area in square feet
OpenPorchSF	numeric	Open porch area in square feet
EnclosedPorch	numeric	Enclosed porch area in square feet
3SsnPorch	numeric	Three season porch area in square feet
ScreenPorch	numeric	Screen porch area in square feet
PoolArea	numeric	Pool area in square feet
PoolQC	text	Pool quality
Fence	text	Fence quality
MiscFeature	text	Miscellaneous feature not covered in other categories
MiscVal	numeric	$Value of miscellaneous feature
MoSold	numeric	Month Sold
YrSold	numeric	Year Sold
SaleType	text	Type of sale
SaleCondition	text	Condition of sale
SalePrice	numeric	The property's sale price in dollars.

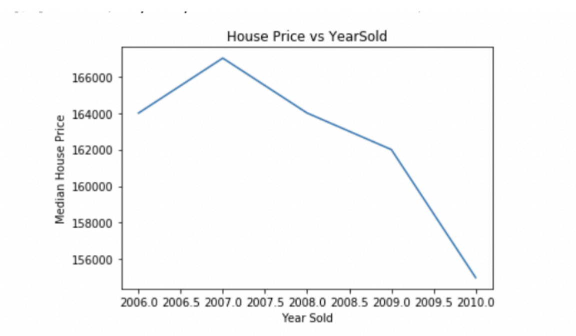
Data Preprocessing

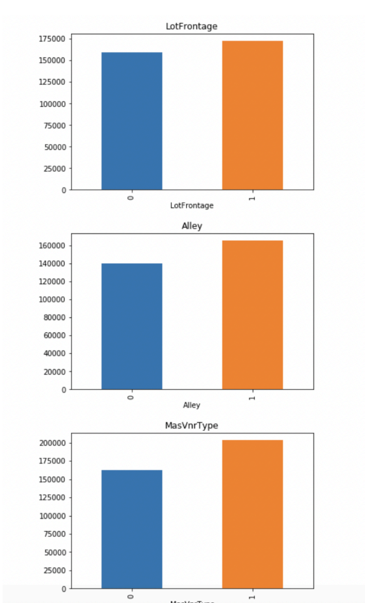
- Handled missing values by replacing with mean for numerical features and mode for categorical features.
- Dropped columns with more than 50% missing values.
- Encoded categorical features using one-hot encoding after combining the training and test sets to ensure consistent encoding.

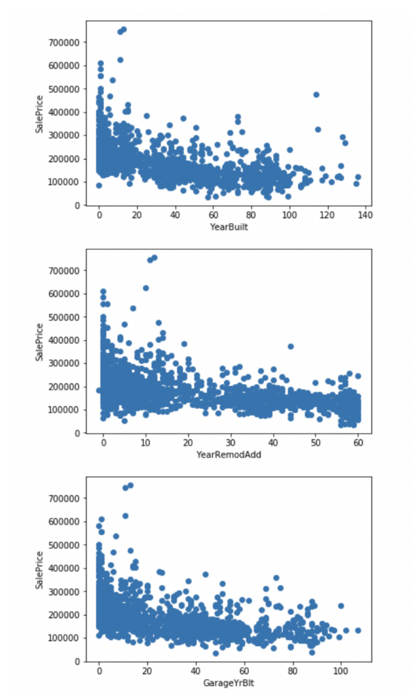
Data Visualisation

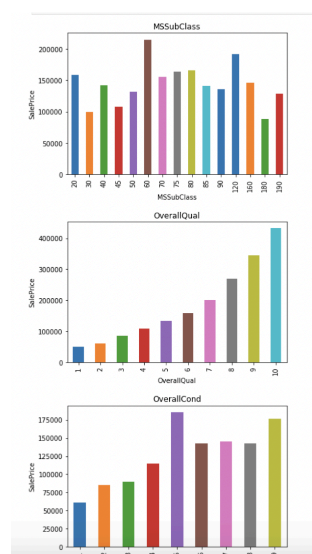
- Tried various regression models like Linear Regression, Decision Tree, Random Forest, and XGBoost.
- Performed hyperparameter optimization using RandomizedSearchCV on the XGBoost model, tuning parameters like n_estimators, max_depth, learning_rate, etc.
- The best XGBoost model was saved as a pickle file for reuse.











Training and Prediction :

Used the optimized XGBoost model to predict the sale prices on the test set.

Prediciton and selecting the Algorithm

In [329…
```
import xgboost
classifier=xgboost.XGBRegressor()
```

In [147…
```
import xgboost
regressor=xgboost.XGBRegressor()
```

In [260…
```
booster=['gbtree','gblinear']
base_score=[0.25,0.5,0.75,1]
```

In [261…
```
## Hyper Parameter Optimization


n_estimators = [100, 500, 900, 1100, 1500]
max_depth = [2, 3, 5, 10, 15]
booster=['gbtree','gblinear']
learning_rate=[0.05,0.1,0.15,0.20]
min_child_weight=[1,2,3,4]

# Define the grid of hyperparameters to search
hyperparameter_grid = {
    'n_estimators': n_estimators,
    'max_depth':max_depth,
    'learning_rate':learning_rate,
    'min_child_weight':min_child_weight,
    'booster':booster,
    'base_score':base_score
    }
```

In [262…
```
# Set up the random search with 4-fold cross validation
random_cv = RandomizedSearchCV(estimator=regressor,
            param_distributions=hyperparameter_grid,
            cv=5, n_iter=50,
            scoring = 'neg_mean_absolute_error',n_jobs = 4,
            verbose = 5,
            return_train_score = True,
            random_state=42)
```

Performance Metrics:

The primary metric used for evaluation is Root Mean Squared Error (RMSE). My initial RMSE was around 0.1415, which ranked me around 2,500 on the leaderboard. After hyperparameter tuning,I was able to improve the RMSE to 0.1349, which improved your ranking to around 2,200.

Plans to further improve the performance by:

- Investigating feature importance and dropping highly correlated features
- Combining the training and test sets to retrain the model and make predictions

Experimental Setup:

Hardware Configuration:
- Laptop with Mac M1 Processor, 8 GB RAM, and SSD for quick
Computations.

Software Stack:
- Programming Language: Python 3.12
- Libraries:
pandas, numpy for data manipulation.
sklearn for machine learning

Deployment:
The project can be deployed locally or on a cloud platform such as Streamlit Share or Heroku

Conclusion & Future Work:

The house price prediction project successfully demonstrated the power of machine learning in real estate valuation, achieving an RMSE of 0.1349 using XGBoost. Future work will focus on developing an interactive web application with real-time data integration, implementing advanced feature engineering techniques, and exploring hybrid machine learning models. The next phase will involve creating a user-friendly frontend using React.js, integrating dynamic APIs for live market data, and expanding the predictive capabilities through ensemble learning and deep learning architectures. By continuously refining the model with advanced techniques like probabilistic predictions, geospatial analysis, and comprehensive feature interactions, we aim to develop a more sophisticated, accurate, and user-centric house price prediction system that can provide valuable insights for buyers, sellers, and real estate professionals.