Project: Predicting Housing Prices

Murali Cheruvu

Project Domain and Data Sources

- Real estate, having more than $50 Billion dollar yearly revenue, is a continued growing industry in United States. With more than 200,000 residential and commercial brokerage firms, there are millions of houses getting sold every year. In recent times, Big Data and Machine Learning have changed the way real estate is getting operated and bringing the importance of data analysis to become major factor in the **decision supporting and making process**.
- The goal of this project is to **predict the sale prices of residential homes listed in the test dataset** as accurately as possible; hence solving a **regression problem**.
- **Kaggle Website** provides the dataset for the housing prices- training and testing, each with 80 features describing various aspects of residential homes in city of Ames, Iowa State; snapshot taken in 2010.
  - There are 1460 rows in the training dataset and 1459 rows in the test dataset.
  - **Out of the 80 variables - 23 are nominal, 23 are ordinal, 14 are discrete, and 20 are continuous.**
  - The nominal variables are related to material, garage, dwelling, and environmental conditions.
  - All the 20 continuous variables are related to the area dimensions.
  - The ordinal variables rate various items within the property.

EDA: Exploratory Data Analysis

We apply **univariate, bivariate and multivariate** analytical techniques; perform various statistical and data visualizations on each feature. The primary goal of exploratory data analysis is to amplify various aspects, such as:

- Numerical and Categorical Analysis
- Feature inter-correlations and correlations with target variable: **Sale Price**
- Missing Value Analysis and Imputation
- Skewed Data Analysis and Data Scaling
- Analyzing the impact of Outliers and fixing them
- Feature Engineering and Ranking
- Good fitting of the model
- Algorithm selection and tuning hyper-parameters for optimal predictions

One-hot Encoding and Feature Engineering

- We have used ordinal and one-hot encoding techniques to convert categorical variables into numerical.
- **One-hot Encoding** converts the category variable into many binary vectors, one new numeric variable for each value in the category.
- Following are a few categorical variables converted to numerical:
  - Lot shape is encoded as: 1 - regular, 2 - Irregular-I, 3 - Irregular-II, 4 - Irregular-III
  - Alley is encoded as: 1 - none, 2 - gravel, 3 - paved
  - All quality variables such as garage quality are encoded as: 0 - none, 1 - poor, 2 - fair 3 - typical 4 - good, 5 - excellent
  - Building type is encoded as: 1 - single-family, 2 - two-family, 3 - duplex, 4 - townhouse end unit, 5 - townhouse
  - Overall quality is encoded as: 1 to 3 - bad, 4 to 6 - average, 7 to 10 – good
- **Feature Engineering** is a technique to analyze all the variables those influence target variable for better predictions. Add new features from existing features, if the new features can add value to the model.

Analysis Outcome

- **Top 5 features** that very **highly correlate with sales price** are: *Over-all-Quality, Ground-Living-Area, Garage-Cars, Garage-Area and Total-Basement-Sq-Ft*.
- **Top 5 features** that **skewed more than 75%** are: *Low-Quality-Fin-Sq-Ft, Ground-Living-Area, Kitchen-Above-Garage, Wood-Deck-Sf, Basement-Half-Bath*.
- **Top 5 features having 90% of the null values**: *Pool-Quality, Misc-Features, Alley, Fence, Fireplace-Quality*.
- Categorical **One-hot encoding** created about **160 new features**.
- **Feature Engineering** added **23 new features** including: *Total-Area, High-Season, Age, Season-Sold, Remodeled*.
- **Top 2 outliers** are: *Ground-Living-Area* and *Garage-Area*; **8 rows** are effected by outliers of these two features.

Cross-Cutting Aspects

Following are some of the aspects that are common to all the algorithms:

- **Underfitting and Overfitting:** Underfitting happens when the model is trivial and does not fit the data properly. Overfitting occurs when the predicted model learns the training dataset including the noise and results negatively impacting the performance and accuracy of the model. Underfitting suffers from low variance but high bias from the predicted model. Overfitting, as expected, exhibits low bias and high variance.
- **Cross Validation**: Cross-validation is a technique to validate the trained model by partitioning the original training dataset into two parts - training and cross validation datasets. **K-fold** cross validation is more effective cross validation method, where the dataset is divided into k subsets, and the holdout method is repeated k times.
- **Model Evaluation:** Regression models use mean absolute error (**MAE**), root mean squared error (**RMSE**), coefficient of determination (**R^2**) and relative scored error (**RSE**) as metrics to verify the accuracy and performance of the model.

ML Pipelines and Components

Two sets of pipelines are used in this project:

- **Pipeline to prepare the training and testing datasets:**
  - Numerical-Feature Pipeline – selecting numerical features, imputation and scaling
  - Categorical-Feature Pipeline – selecting categorical features, categorical imputation and one-hot encoding
  - Feature Union – to combine the numerical and categorical pipeline components
  - Feature Engineering

- **Pipeline to run the ML algorithms on the prepared and cleaned datasets:**
  - Ridge, Lasso, SVM, Random Forest and XGBoost

Workflow

- Apply ML pipeline to clean, scale, encode and apply feature engineering on the training and testing datasets separately; make sure one dataset will not impact the other during the preparation process.
- Make sure all the features those have been present in the training dataset are also there in the test dataset as ML algorithms expect fit and predict methods apply on the same set of features.
- Apply univariate feature to select top 20% best features based on their statistical significance using metrics like *f_regression, f_classif* and *chi-square* statistics through Select Percentile modeling
- Create cross-validation dataset from training dataset in the ratio of **70-30**
- Apply ML pipeline of algorithms: **Ridge, Lasso, SVM, Random Forest and XGB** through K-Fold cross-validation and collect various performance metrics – MAE, MSE, RMSE and R^2
- Compare the top 2 performing algorithms: **Random Forest and XGB** with the third performing algorithm SVM as baseline; do the **statistical significance** to prove the same.
- Tune the hyper-parameters of both Random Forest and XGB models using **Grid Search** and fit/predict the training / cross-validation datasets and compare them with the actual results; then predict the sale prices of test dataset.
- **Ensemble** Random Forest and XGB to get better predictions and submit the predictions to Kaggle to get better score.

Conclusions

- TOP 3 performing algorithms are XGB, Random Forest and SVM
- Ensembling top 2, XGB and Random Forest, gave better performance
- Extract the feature importance of XGB to make sure they are meaningful, and as per our analysis

Metrics: **Kaggle Scores** are based on **R^2 Metric**

Algorithm	MAE	MSE	RMSE	R^2	Kaggle Score
Random Forest	0.1196	0.0316	0.1777	0.802	0.14607
XG Boost	0.033	0.0020	0.0449	0.987	0.13102
SVM	0.071	0.009	0.096	0.942	
Ridge	0.088	0.015	0.123	0.096	
Lasso	0.095	0.018	0.134	0.887	
Ensemble (top 2)					0.12600