Group 30: Ziming Wang

House Price Prediction Using Regression Techniques

Introduction

Predicting house prices is a challenging and important task in the real estate industry. It is essential to have a model that can accurately predict house prices to help buyers, sellers, and real estate agents make informed decisions. There are several challenges involved in predicting house prices. One of the main challenges is that the real estate market is complex, and many factors contribute to the price of a property. These factors include the location of the property, the size and condition of the house, the local real estate market, and the overall economic conditions.

The main techniques in this project are clustering algorithms and regression. Clustering algorithms can help identify attribute groups or clusters with similar characteristics. This helps realtors understand the characteristics of different neighborhoods and the types of properties needed in those areas. For example, clustering can be used to group houses by location, size, and price range. On the other hand, regression techniques can be used to predict the price of a property based on its characteristics. Regression models can analyze data sets of attributes and their corresponding prices to determine the most important characteristics affecting house prices. Both techniques can be used to predict the price of a new property based on its features. In this project we want to explore which technology performs best in this housing price prediction.

Data Collection and Exploration

We obtained the house price dataset from the Ames Housing dataset and Kaggle. This dataset contains information on various aspects of each house. For example, Size of garage in car capacity, Number of fireplaces, etc. This training data set has a total of 1460 rows and 81 columns. In this dataset, there are both numerical and categorical features, and some features contain missing values. The dataset was collected between 2006 and 2010, which means that the models built on this dataset may not accurately predict the current market trends. This dataset may have bias because this dataset only includes properties located in Ames, Iowa. Therefore, the models built on this dataset may not accurately predict the prices of properties located in other cities or countries. However, this dataset is good enough for our project, the dataset contains many observations and features with a wide variety of information about the properties.

Import the Data: The first step in preprocessing any dataset is to import the data into our programming environment. The Ames Housing dataset is provided in two CSV files: train.csv and test.csv.

Cleaning the Data: Once we have imported the data, the next step is to clean it. This involves identifying and addressing missing values, handling outliers, and dealing with any other anomalies in the data. One common issue in real-world datasets is missing values. These are typically represented as “NaN” or “None” values in pandas dataframes. To handle missing values in the Ames Housing dataset, we can use the following steps: 1. Identify the columns with missing values. 2. Decide how to handle the missing values for each column. Here I choose to use two different approaches: dropping the rows with missing values and replacing the missing values with a default value like the mean or median. After testing the dataset when we perform both approaches, we found that using the default value is much better. That is because the dataset has 80 feature columns and some of them have missing values. If we delete them, we will lose many rows of our training dataset.

Another common issue in datasets is outliers, which are extreme values that do not follow the general pattern of the data. Outliers can skew our model and make it less accurate, so it's important to identify and handle them. In this step, we still have two choices which are dropping the rows or replacing the outlier with a median value. After testing, we found that there are no clear changes to our dataset. The outliers are not too many, so we decided to delete the outliers.

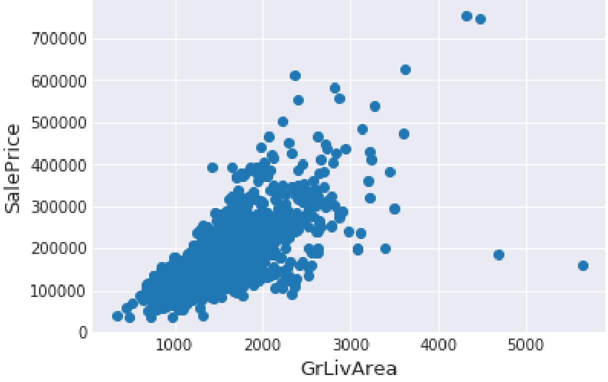
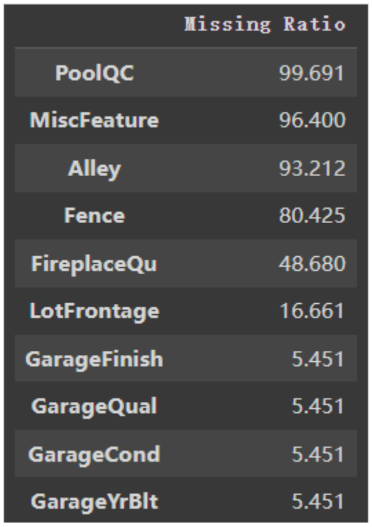


Figure 1: Outliers



	Missing Ratio
PoolQC	99.691
MiscFeature	96.400
Alley	93.212
Fence	80.425
FireplaceQu	48.680
LotFrontage	16.661
GarageFinish	5.451
GarageQual	5.451
GarageCond	5.451
GarageYrBlt	5.451

Figure 2: Missing Values

Feature Selection: Because our dataset has 81 columns with the target label column. We need a way to reduce the dimensionality of the dataset and improve the performance and efficiency of subsequent training. Therefore, we need a method to measure the importance of each feature. In this step, I chose the correlation-based feature selection which involves selecting features that are highly correlated with the target variable while discarding weakly or irrelevant features. This approach is easy to use. After obtaining each correlation for each column, we found that the in this dataset, some features such as “OverallQual”, “GrLivArea”, “GarageCars”, “GarageArea”, “TotalBsmtSF”, “1stFlrSF”, “FullBath”, and “YearBuilt” are likely to have a strong correlation with the target variable. For these columns we kept them as the important aspects of sales housing price.

Transforming categorical variables to numerical: In the Ames Housing dataset, there are many categorical variables such as MSZoning (general zoning classification), Street (type of road access), and Utilities (type of utilities available). These variables provide important information about the property, but most machine learning algorithms require numerical inputs. One method we used is one-hot encoding which creates a binary variable for each category in a categorical variable. By transforming categorical variables to numerical, we can ensure that all variables in the dataset are compatible with the machine learning algorithm being used.

Finally, we got the train.csv with 1020 rows and 30 columns. By removing these useless or unimportant data, we can improve the performance of the model by reducing overfitting and improving its ability to generalize to new data.

Clustering Algorithms

Halfway through the course of this semester, we studied clustering algorithms and learned one of the most popular algorithms, k-means algorithm. K-means is a clustering algorithm that is used to partition a given dataset into K clusters. The algorithm works by grouping together data points that are similar to each other based on their proximity to a centroid. We think k-means can be implemented to our In the housing price dataset, the characteristics of houses with higher prices are not very different. These features can be aggregated well and can predict the price of other input houses. For these reasons, we tried the k-means algorithm.

However, after testing we observed that our dataset has a lot of features. Our dataset contains a mix of continuous, categorical and ordinal features. After performing the k-means algorithm, we found that k-means can be used to deal with continuous numerical data very well. But may not handle mixed data types well. Although we will convert some categorical variables to numerical. Besides this, K-means assumes that the relationships between the features are linear. However, in the Ames Housing dataset, there may be non-linear relationships between the features that could be affecting the clustering results, which means that they cannot be separated by straight lines or hyperplanes.

Then my team members tried other clustering algorithms, such as DBSCAN algorithm. DBSCAN algorithm is a density-based clustering algorithm. The DBSCAN-based algorithm works best when the clusters have similar densities. However, in the Ames Housing dataset, clusters can have very different densities depending on the type of housing (for example, single-family versus multifamily). The Ames Housing dataset has a relatively large number of features, which makes it difficult for DBSCAN to identify meaningful clusters. DBSCAN is dimensionality sensitive, which means its performance degrades as the number of features increases.

Regression

We learned from the classroom and network resources that housing price prediction is a regression problem. Therefore, we gave up continuing to use clustering algorithms and began to study regression.

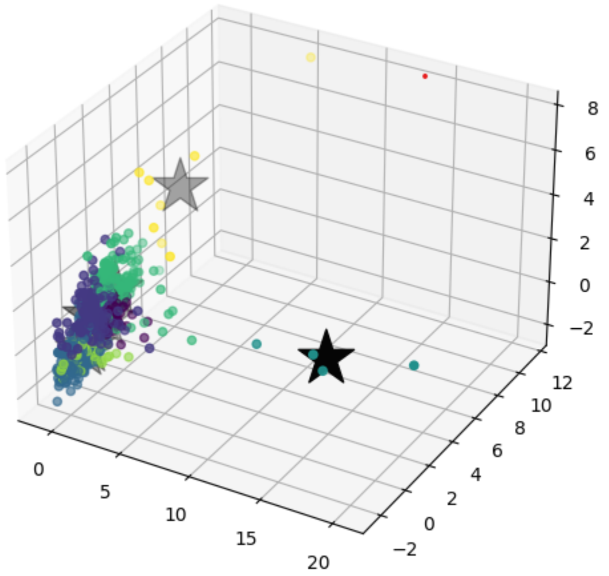


Figure 3: K-means Clustering

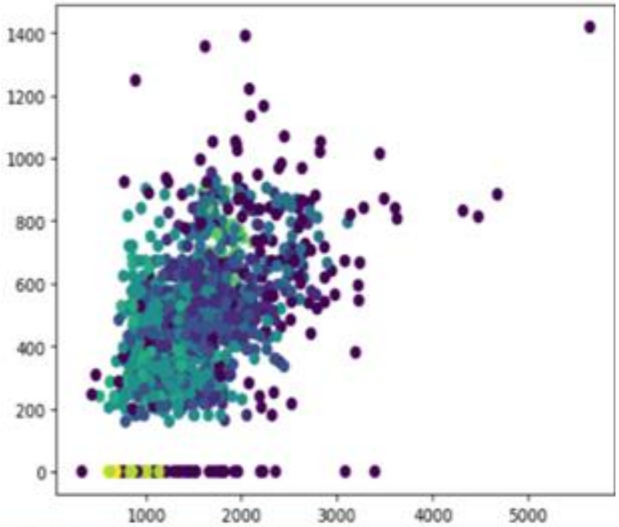


Figure 1DBSCAN clusters graph

Regression Algorithm	Linear Regression	L1	L2	Random Forest Regression	Gradient Boosting Regression
RSME	≈ $33327	≈ $37120	≈ $25627	≈ $20112	≈ $11314

We tried many regression algorithms. Such as linear regression, L1 and L2 regularization, etc. We evaluate the performance of each algorithm by using root mean square error (RMSE). Finally we found that the Gradient Boosting Regression algorithm has the lowest RMSE. The Ames Housing dataset contains many nonlinear relationships between predictor and target variables. Gradient Boosting Regression can model complex nonlinear relationships between features and target variables, making it well suited for this dataset. Also, the Ames Housing dataset contains missing data, which can be a challenge for some machine learning algorithms. Gradient Boosting Regression can handle missing data by feeding in values or ignoring missing data during training.

Conclusion and Discussion

In this project, we learned how to preprocess data, which involves cleaning, transforming, and preparing the dataset for analysis. This included dealing with missing data, encoding categorical features, scaling numerical features, and removing outliers. I learned how clustering and regression algorithms work and that they can be used in many applications, such as sales forecasting, risk analysis, and demand forecasting. Both clustering and regression are important in data mining because they help make sense of complex data and reveal insights that can drive business decisions.