Data*6100 Project 1 – House Price Prediction on Ames Housing Dataset
Under the guidance of Prof. Mihai Nica
Student: Atharva Vichare

Executive Summary:

In this DATA*6100 project, we will be using Ames Housing Dataset to predict House Sale Price on the test dataset and minimise the RMSE. I have used Linear Regression Model to predict the sale prices. To improve our prediction, we will perform data pre-processing and manipulation steps like handling outliers, filling missing data, ordinal and one hot encoding etc to figure out how we can train the best performing Linear Regression Model. I will also take you through why I decided to do some pre-processing and cleaning steps.

Data Pre-processing:

- The first step is reading the dataset from the csv file we downloaded. We will use the read_csv() function from Pandas Python package and then join both the datasets row wise using concat() function.
- Drop unnecessary columns like “Unnamed: 0” & “ID”.
- Plot histograms and see that most of our data contains categorical values which we need to deal with, they are some missing values in the dataset as well as most of our data is skewed heavily.
- Data Imputation with mean median or mode for numerical columns
- Filling NA values in categorical columns like:
  BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, GarageType, GarageFinish, GarageQual, GarageCond, PoolQC, MiscFeature, Fence, Alley, MasVnrType, FireplaceQu.
- Fill "MasVnrArea" with 0 value because having null value in this column indicated "MasVnrType" for that particular house.
- Fill GarageYrBlt" we will fill the null values with 0 because it indicates the house has no Garage.
- Impute null value in "Electrical" column with mode because it a categorical type column.
- 259 null values in in LotFrontage column will be filled with mean based on Neighborhood because it represents part of street connected to the property and it varies from neighborhood to neighborhood.
- Columns like ExterQual, ExterCond, HeatingQC, KitchenQual, BsmtQual,BsmtCond, GarageQual, GarageCond, FireplaceQu, BsmtFinType1, BsmtFinType2, BsmtExposure which contain natural order so we will do ordinal encoding in them for our model to understand the features better.
- Converting columns like MoSold, YrSold & MsSubclass into categorical type.
- One hot encoding for all the remaining categorical columns.
- Deleting Multi co related features to improve our model performance and decrease feature space.
- Handling outliers with LOF (Local Outlier Factor) algorithm

Modelling and Model Tunning:

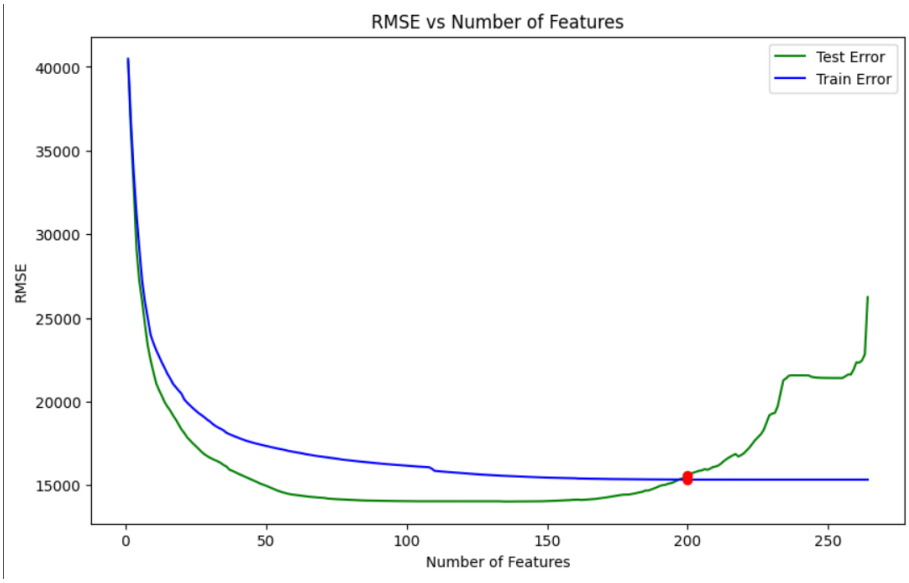
After pre-processing and splitting the dataset back into training and testing I performed Forward Feature selection to get the best features with minimum RMSE. It starts with 0 features, adds a feature one by one for each feature and test the RMSE on the train set itself. Once the loop is over, it keeps adding one one feature to minimise the RMSE the most, it keeps on repeating the process on and on until it feels that the RMSE is not decreasing significantly now. We got 135 features as final features with minimum RMSE of 14025.34.

I have selected Linear Regression model taught in class to predict SalePrice. A linear regression model is a supervised learning algorithm in machine learning that aims to model the relationship between the target

variable (i.e. SalePrice in our case) and one or more independent variables (i.e. 135 features selected through forward selection) by fitting a linear equation to the data.

Demonstration of Overfitting and Underfitting:

In machine learning, overfitting occurs when an algorithm fits too closely or even exactly to its training data, resulting in a model that can't make accurate predictions or conclusions from any data other than the training data.



From the graphs we can say that RMSE decreases steadily as the number of features increases, indicating that the model fits the training data better with more features. This can lead to overfitting if the model becomes too complex. As more features are added, indicating an improvement in model performance. However, after reaching a certain point the test error starts to increase, suggesting overfitting. Underfitting in machine learning occurs when a model is too simple to capture the underlying patterns in the data. This results in poor performance on both the training and test datasets because the model cannot generalize well to new data. Underfitting usually happens when the model has insufficient complexity or when it fails to learn the relationships in the data properly. From the plot , we can see that when number of features is less , the model is underfitted which results in high error in both test and test dataset.

The U-shaped curve shown in the graph represents how the number of features in the model impact the RMSE for the test dataset, which indicates the error in predicting housing prices. At the left end of the curve, where the number of features is very low, the model is too simple to effectively learn the underlying relationships in the data. This results in high test error due to high bias, meaning the model fails to capture important patterns and trends, leading to poor performance on both training and test data. After a few more features the test error starts to rise again, leading to overfitting.

Final Model Prediction:

So, the final model contains 135 features which give minimum RMSE on test data

The TEST RMSE submitted on leaderboard is: **21195**