**CEREBRO – Beginner’s Hypothesis**

**Name:** Priyanshi Jaiswal

**Overview**

1. Importing Packages
2. Loading and Inspecting data
3. Feature Engineering
4. Creating, Training, Evaluating, Validating, and Testing ML Models
5. Submission

Importing Packages:

* I used the same modules as I would use for any problem working with data. I imported numpy and pandas to work with numbers and data, and seaborn and matplotlib to visualize data.

Loading and Inspecting:

* With various Pandas functions, I loaded my training and test data set as well as inspected it to get an idea of the data I was working with.
* I did some analysis on the target variable, I used sns.distplot to visualise distribution of the target variable, I also used probability plot from sns to compare the sample and theoretical quantiles. The target variable was right skewed. As (linear) models love normally distributed data , I needed to transform the variable and make it more normally distributed. So I did log transformation of the target variable.

Feature Engineering:

* Missing values: To fix these first I took the percentage missing data in the features, than to visualise I plotted it on the graph, I eliminated the features having missing data more than 50% as it was not significant for prediction, for the rest of the features I replaced the null values by mode of the values present in the respective categorical features.
* Encoding the data: I encoded the categorical features using label encoder imported from sklearn library.
* Data correlation: To check the correlation between features I plotted a heatmap with correlation values.

Creating, Training, Evaluating, Validating, and Testing ML Models:

* Now that I had pre-processed and explored the data, I had a much better understanding of the type of data. So, I began to build and test different models for regression to predict the Price of each house. I imported these models, train them, and evaluated them. I used the R^2 score as well as the RMSE to evaluate the model performance.
* PCA: I used PCA as It reduces the dimension of your data with the aim of retaining as much information as possible. Before applying PCA I had to convert all the data into a single scale. I used Standard Scalar method to scale the data.
* Linear Regression: I fit the training data transformed by PCA in Linear regression, I split the training data in train and test (validation data), I evaluated the rmse score on the basis of y test and output predicted by the model.
* Decision Tree: I used decision tree regressor along with GridSearchcv to find the hyperparameters, such as n\_estimators and depth, in order to get the best result.
* Random Forest Regressor: I used Random Forest Regressor along with GridSearchcv to find the hyperparameters, such as n\_estimators and depth, in order to get the best result.
* I tried to improve the performance by using boosting techniques such as ADABOOST and GradientBoosting.
* I got the least rmse with gradient boosting, so evaluated my actual test data with it.