1. Import numpy, pandas, seaborn, matplotlib
2. Import data as “training” and “X\_test”, print “training” ‘s columns, shape. Print X\_test shape
   * Check training: info
3. Write function to create scrollable table “create\_scrollable\_table”
4. Create numerical dataset as “df\_num”. Use “create\_scrollable\_table” on statistics summary of df\_num (transpose) 🡺 display
   * Check df\_num: info, shape
5. Create categorical dataset as “df\_cat”. Use “create\_scrollable\_table” on statistics summary of df\_cat (transpose) 🡺 display
   * Check df\_cat: info, shape
6. Check null values on each column. Use “create\_scrollable\_table” to display
7. Check percentage of null values on each column. Use “create\_scrollable\_table” to display
   * Check those with null percentage > 5%
8. Show rows with missing values. Use “create\_scrollable\_table” to display
9. Create a “SalePrice Distribution” graph, consisting of:
   * A histogram of “SalePrice”
   * A normal distribution line
10. Create a “Q-Q plot”, consisting of:
    * A Q-Q plot
    * Best line fit
11. Write function to show value on bar graphs.
12. Make bar graphs showing:
    * Distribution of building types
    * Distribution of dwelling types and their relation to average sale prices
13. Make bar graphs showing the impact of zoning on sale prices.
14. Make bar graphs showing the impact of street and alley on sale prices.
15. Make bar graphs showing the
    * Average sale price by shape.
    * Average sale price by contour.
16. Make a scatter graph showing the relationship between SalePrice and PropertyAge (needs calculation)
17. Make a scatter graph showing the relationship between SalePrice and Living Area
    * Also, calculate the correlation between SalePrice and Living Area
18. Make a box plot showing SalePrice trends over the years.

PREPROCESSING PIPELINE:

1. Import: ColumnTransformer, Pipeline, SimpleImputer, StandardScaler, OneHotEncoder
2. Define transformers for numerical and categorical columns.
3. Use ColumnTransformer to apply numerical\_transformer for numerical columns and categorical\_transformer for categorical columns.
4. Create preprocessor Pipeline. Create X\_train, y\_train. Apply pipeline on those training sets to get “X\_train\_preprocessed”, “X\_test\_preprocessed”.

MODEL BUILDING (Preprocessor only)

1. Import: LinearRegression, RandomForestRegressor, XGBRegressor, GridSearchCV, KFold
2. Make a “models” dictionary containing LinearRegression, RandomForestRegressor, XGBRegressor
3. Define param\_grids. Define cv=KFold
4. Run GridSearchCV/ Fit training data on each mode (X\_train\_preprocessed). Print best\_params and best\_score of each model.

MODEL BUILDING (PCA)

1. Import PCA
2. Use PCA to decrease dataset’s dimension. Put the PCA with reduced dimension to preprocessor Pipeline. Apply pipeline on those training sets to get “X\_train\_preprocessed\_pca”
3. Import: LinearRegression, RandomForestRegressor, XGBRegressor, GridSearchCV, KFold
4. Make a “models” dictionary containing LinearRegression, RandomForestRegressor, XGBRegressor
5. Define param\_grids. Define cv=KFold
6. Run GridSearchCV/ Fit training data on each mode (X\_train\_preprocessed\_pca). Print best\_params and best\_score of each model.

FEATURE ENGINEERING (FE)

1. Use “create\_scrollable\_table” to display some specific columns of X\_train, called “var\_explore”
2. Define feature engineering function:
   * Create PropertyAge column
   * Create TotalSF column
   * Create TotalBath column
   * Create HasRemodeled column
   * Create Has2ndFloor column
   * Create HasGarage column
   * Change YrSold column to object type, as YrSold\_cat
   * Change MoSold column to object type, as MoSold\_cat
   * Change YrBuilt column to object type, as YrBuilt\_cat
   * Change MSSubClass column to object type, as MSSubclass\_cat
3. Import FunctionTransformer and feed the feature engineering function to it.
4. Add ‘HasRemodeled’, ‘Has2ndFloor’, ‘HasGarage’ to categorical columns.
5. Add ‘PropertyAge’, ‘TotalSF’, ‘TotalBath’, ‘YrSold\_cat’, ‘MoSold\_cat’, ‘YearBuilt\_cat’, ‘MSSubClass\_cat’ to numerical columns.
6. Use ColumnTransformer to apply numerical\_transformer for numerical columns and categorical\_transformer for categorical columns.
7. Create preprocessor Pipeline. Create X\_train, y\_train. Apply pipeline on those training sets to get “X\_train\_preprocessed\_FE”, “X\_test\_preprocessed\_FE”

MODEL BUILDING (FE + PCA)

1. Import: LinearRegression, RandomForestRegressor, XGBRegressor, GridSearchCV, KFold
2. Make a “models” dictionary containing LinearRegression, RandomForestRegressor, XGBRegressor
3. Define param\_grids. Define cv=KFold
4. Run GridSearchCV/ Fit training data on each mode (X\_train\_preprocessed\_FE). Print best\_params and best\_score of each model.
5. Choose the best model and predict “y\_pred”. Export to csv file.