PROBLEM STATEMENT 1 :

HOUSE PRICES

AIM :

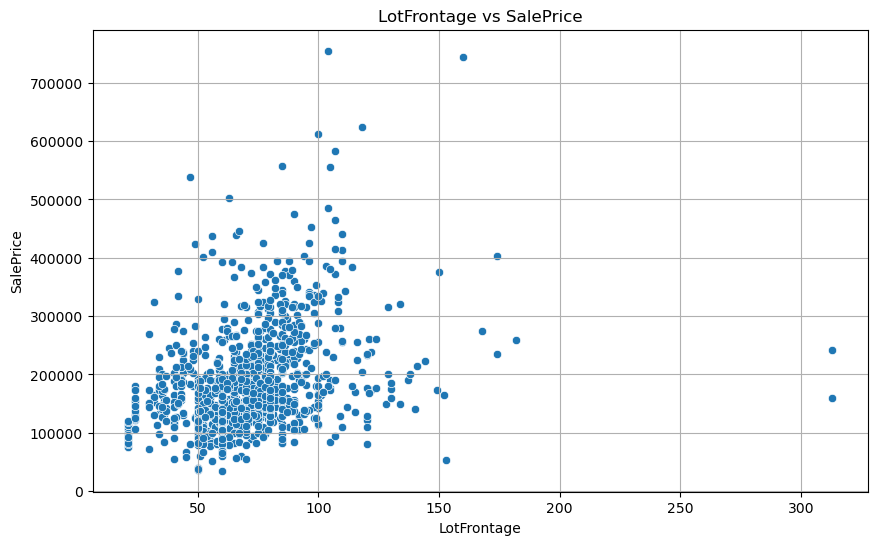
To predict the sales price for each Id in the test set using 79 explanatory variables describing (almost) every aspect of residential homes.

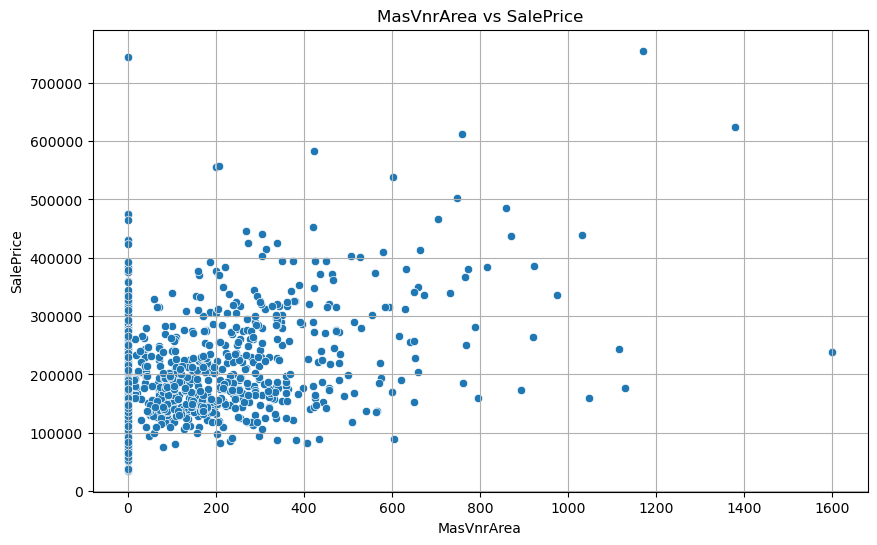
**1. DATA PROVIDED :**

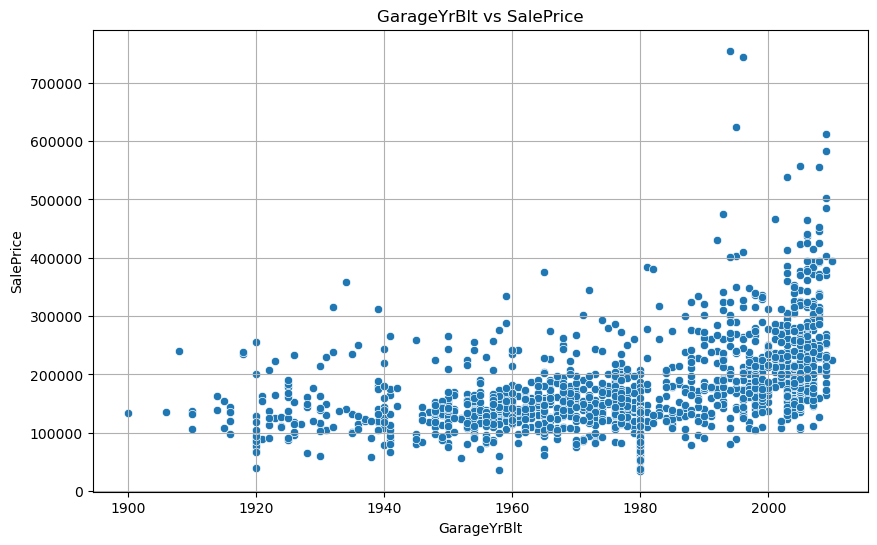
* train.csv – contains the labelled training set with the target variable (Sales Price)
* test.csv – contains unlabelled test set
* data\_description.txt - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here
* sample\_submission.csv - a benchmark submission from a linear regression on year and month of sale, lot square footage, and number of bedrooms

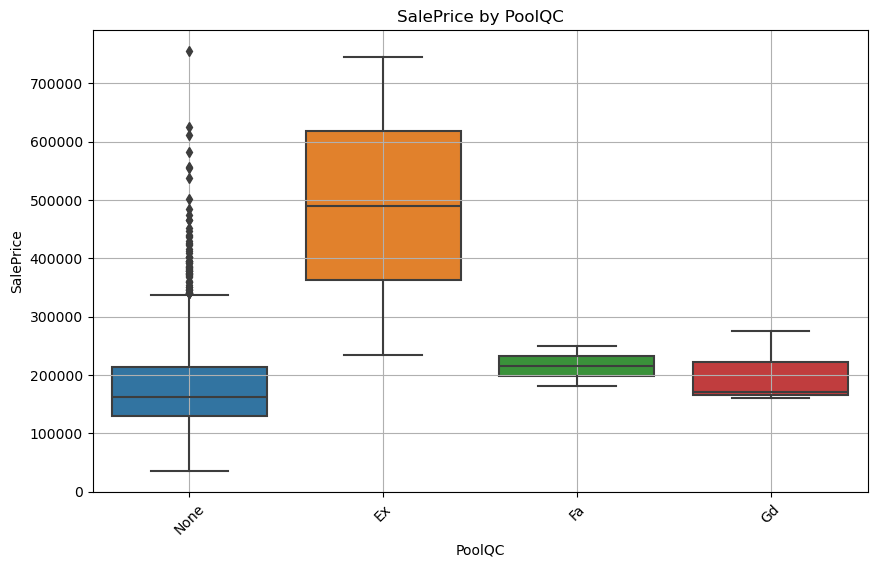
**2. DATA PREPROCESSING :**

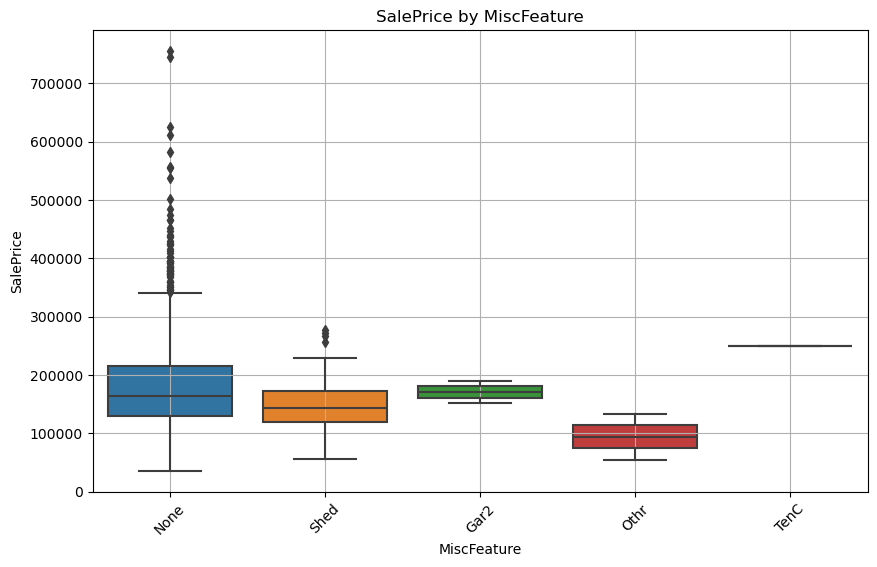
* **Data Loading:**  
  We loaded the train and test datasets, with the target variable being SalePrice from the training data.
* **Categorical Features**:
* Many categorical features had missing values, and these were filled with the most appropriate category, often 'None' where the absence of a feature was meaningful (e.g., no pool or no fireplace).
* PoolQC, MiscFeature, Alley, Fence, MasVnrType, FireplaceQu, GarageCond, GarageType, GarageFinish, GarageQual, BsmtFinType2, BsmtExposure, BsmtQual, BsmtCond, BsmtFinType1: Filled with 'None' indicating the absence of these features.
* **Numerical Features**: For numerical columns with missing values, we used either the median or mode to impute the missing values, ensuring that outliers did not overly influence the imputation.
  + **GarageYrBlt**: Filled with the median value to maintain continuity in the dataset.
  + **MasVnrArea**: Filled with the median value, as it's a numerical feature representing the area of the masonry veneer.
  + **Electrical:** Filled with the mode (most common category), as this is typically a categorical feature with some missing entries.
* **Data Visualization :** To gain insights into the relationship between features and the target variable (SalePrice), we plotted the data for both **numerical** and **categorical features**. Some graphs are as shown below.

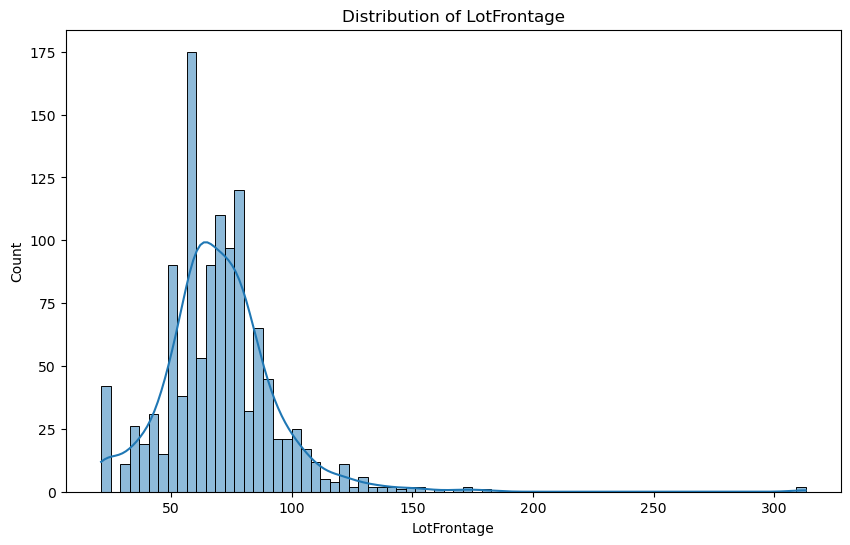


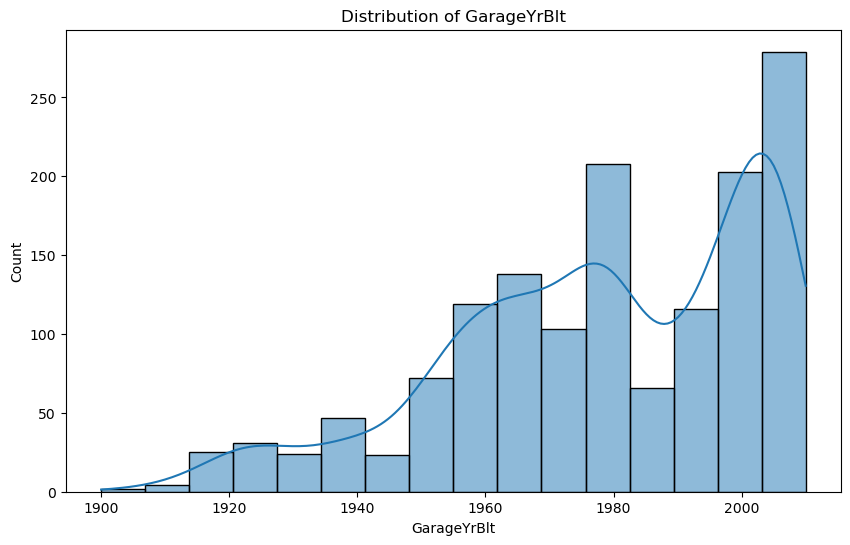












* **Feature Engineering:**
  + Removed the Id column, as it was not relevant for predictions.
  + Created new interaction features such as TotalSF (sum of basement and floor areas) and TotalBath (sum of full and half bathrooms).
  + Generated polynomial features for key variables (GrLivArea and OverallQual) to capture non-linear relationships between these features and the target variable.
* **Feature Encoding and Data Preparation:**
* **Binary Encoding**: Applied to binary features like CentralAir, which only have two categories. Binary encoding is simple, efficient, and directly interpretable by the model.
* **Ordinal Encoding**: Suitable for ordinal features with a clear ranking or order. By encoding ordinal features this way, the model can utilize the natural hierarchy within the categories, which can be important for making predictions.
* **One-Hot Encoding**: Used for nominal features, which have no inherent order. This method prevents the model from mistakenly assuming any sort of ordinal relationship between the categories, which could lead to erroneous predictions.
* **Standardization:**
  + Applied **StandardScaler** to standardize the features, ensuring that they are on a similar scale, which helps with model convergence, especially for gradient boosting methods.

**3. MODEL SELECTION :**

* Given the complexity of the dataset, we opted for **XGBoost**, a highly efficient and scalable tree-based model. XGBoost offers robust handling of various feature types, regularization options to avoid overfitting, and built-in handling of missing data.
* **XGBoostRegressor:** An optimized gradient boosting model, ideal for structured data like the one we are working with.

**4. HYPERPARAMETER TUNING :**

* We used **RandomizedSearchCV** to optimize the hyperparameters of the XGBoost model. RandomizedSearchCV is more efficient than GridSearchCV when searching large hyperparameter spaces, as it samples a fixed number of hyperparameter combinations.
* **Parameters Tuned:**
  + learning\_rate: Step size during updates (tested values: [0.01, 0.05, 0.1]).
  + n\_estimators: Number of boosting rounds/trees (tested values: [500, 1000, 2000]).
  + max\_depth: Maximum depth of each tree (tested values: [3, 4, 5]).
  + gamma: Minimum loss reduction to partition a leaf (tested values: [0, 0.1, 0.2]).
  + reg\_alpha: L1 regularization term to prevent overfitting (tested values: [0, 0.0001, 0.001]).
  + subsample: Fraction of samples used per tree (tested values: [0.7, 0.8, 0.9]).
  + colsample\_bytree: Fraction of features used per tree (tested values: [0.7, 0.8, 0.9]) .
  + min\_child\_weight: Fraction of features used per tree (tested values :[1,5,15,200]).
  + 'lambda': Fraction of features used per tree (tested values : [1,2,3])
* **Cross-Validation:**  
  We performed 5-fold cross-validation during RandomizedSearchCV to evaluate model performance effectively.

**5. MODEL TRAINING and EVALUATION :**

* **Training the Model:**  
  After identifying the best hyperparameters, we trained the optimized XGBoost model on the training dataset.
* We split our data into two parts, ‘train’ and ‘test’ with ‘test’ having 20% of the data. This was done to train the model on the ‘train’ part and check its working and make predictions on the ‘test’ part
* **Performance Metrics:**  
  The model was evaluated using the following metrics:
  + **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values.
  + **Root Mean Squared Error (RMSE):** Provides an interpretable metric in the same units as the target variable.
  + **Mean Absolute Error (MAE):** Measures the average magnitude of prediction errors.
  + **R-squared (R²):** Indicates how well the model explains the variance in the target variable.

**6. FEATURE IMPORTANCE :**

The XGBoost model provided feature importance scores, helping us understand the most influential factors in the model’s predictions.

* **Key Influential Features:**
  + OverallQual: Reflects the overall quality of the house.
  + GrLivArea: Above-ground living area.
  + TotalSF: Total square footage.

**7**. **RESULT :**

* **Model Performance on Test Data:**
* **MSE:** **0.017**
* **RMSE: 0.13**
* **MAE:** **0.088**
* **R² Score:** **0.90**

**8. CONCLUSION :**

By leveraging XGBoost with RandomizedSearchCV, our model achieved strong results in key performance metrics, and feature importance analysis highlighted key drivers of house prices**.**