

PREDICTING HOUSE PRICE USING MACHINE LEARNING

Phase-1 Document Submission

TEAM MEMBERS: Sivaranjini J {311521106092}

Prithisha V {311521106073}

Swetha S {311521106103}

Yamini R {311521106115}

PROJECT TITLE: House Price Prediction



OBJECTIVE:

The objective of the predicting house price project is to develop a machine learning model that can accurately predict the price of a house in a specific geographic area. The model will be trained on a dataset of historical house sale prices and other relevant factors, such as property characteristics, location, and market conditions.

1. Data Source:

Empathize: Understand the needs of real estate stakeholders. What are their main sources of data for house price prediction? What data would be most useful to them?

- **Ideate:** Brainstorm different data sources that could be used to predict house prices. This could include public datasets, such as Zillow's Home Value Index, or private datasets that are collected by real estate agents or investors.

DatasetLink: <https://www.kaggle.com/datasets/vedavyasv/usa-housing>

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

2.Data Pre-processing:

- **Empathize:** Understand the challenges of cleaning and pre-processing house price data. What types of errors and inconsistencies are common? How can these errors be identified and corrected?
- **Ideate:** Brainstorm different ways to clean and pre-process house price data. This could involve handling missing values, converting categorical features into numerical representations, and normalizing the data.

a) Duplicate Removal: Duplicate rows can introduce bias into the model. We will identify and remove Duplicates, typically by sorting the dataset based on a unique identifier (e.g., property ID) and then eliminating consecutive rows with the same identifier.

b) Handling Missing Values: Missing data is common and needs to be addressed. We will utilize suitable methods, Such as:

- **Mean Imputation:** Replace missing values with the mean of the feature for the remaining rows. This is appropriate for numerical features.

- **Median Imputation:** If data contains outliers, median imputation can be more robust as it is less sensitive to extreme values.

c) **Categorical Variable Encoding:** Categorical variables, such as property type or location, need to be converted into numerical form so that machine learning models can process them. Two common approaches include:

- **One-Hot Encoding:** Create binary columns for each category, representing the presence or absence of that category.
- **Label Encoding:** Assign a unique integer to each category, preserving the ordinal relationship if applicable.

d) **Data Normalization:** To ensure that all features are on a consistent scale, normalization techniques can be applied. This includes:

- **Standardization:** Scaling features to have a mean of 0 and a standard deviation of 1.
- **Min-Max Scaling:** Scaling features to a specified range (e.g., 0 to 1)



PYTHON PROGRAM:

```
# Import necessary libraries

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler

# Load the dataset (replace 'house_data.csv' with your dataset file)

data = pd.read_csv('E:/USA_Housing.csv')

# Display the first few rows of the dataset to get an overview

print("Dataset Preview:")

print(data.head())

# Data Preprocessing

# 1. Handle Missing Values

# Let's fill missing values in numeric columns with the mean and in
categorical columns
with the most frequent value.

numeric_cols = data.select_dtypes(include='number').columns

categorical_cols = data.select_dtypes(exclude='number').columns

imputer_numeric = SimpleImputer(strategy='mean')

imputer_categorical = SimpleImputer(strategy='most_frequent')
```

```
data[numeric_cols]=imputer_numeric.fit_transform(data[numeric_cols])
```

```
data[categorical_cols]=imputer_categorical.fit_transform(data[categorical_cols])
```

2. Convert Categorical Features to Numerical

We'll use Label Encoding for simplicity here. You can also use one-hot encoding for

nominal categorical features.

```
label_encoder = LabelEncoder()
```

```
for col in categorical_cols:
```

```
data[col] = label_encoder.fit_transform(data[col])
```

3. Split Data into Features (X) and Target (y)

```
X = data.drop(columns=['Price']) # Features
```

```
y = data['Price'] # Target
```

4. Normalize the Data

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

Split data into training and testing sets (adjust test_size as needed)

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,  
test_size=0.2,
```

```
random_state=42)
```

Display the preprocessed data

```
print("\nPreprocessed Data:")
```

```
print(X_train[:5]) # Display first 5 rows of preprocessed features  
print(y_train[:5]) # Display first 5 rows of target values
```

OUTPUT:

Dataset Preview:

Avg. Area Income Avg. Area House Age Avg. Area Number of
Rooms \

0 79545.458574 5.682861 7.009188

1 79248.642455 6.002900 6.730821

2 61287.067179 5.865890 8.512727

3 63345.240046 7.188236 5.586729

4 59982.197226 5.040555 7.839388

Avg. Area Number of Bedrooms Area Population Price \

0 4.09 23086.800503 1.059034e+06

1 3.09 40173.072174 1.505891e+06

2 5.13 36882.159400 1.058988e+06

3 3.26 34310.242831 1.260617e+06

4 4.23 26354.109472 6.309435e+05

Address

0 208 Michael Ferry Apt. 674\nLaurabury, NE 3701...

1 188 Johnson Views Suite 079\nLake Kathleen, CA...

2 9127 Elizabeth Stravenue\nDanielstown, WI 06482...

3 USS Barnett\nFPO AP 44820

4 USNS Raymond\nFPO AE 09386

Preprocessed Data:

[[-0.19105816 -0.13226994 -0.13969293 0.12047677 -0.83757985 -
1.00562872]

[-1.39450169 0.42786736 0.79541275 -0.55212509 1.15729018
1.61946754]

[-0.35137865 0.46394489 1.70199509 0.03133676 -0.32671213
1.63886651]

[-0.13944143 0.1104872 0.22289331 -0.75471601 -0.90401197 -
1.54810704]

[0.62516685 2.20969666 0.42984356 -0.45488144 0.12566216
0.98830821]]

4227 1.094880e+06

4676 1.300389e+06

800 1.382172e+06

3671 1.027428e+06

4193 1.562887e+06

Name: Price, dtype: float64

3.Feature selection :

- **Empathize:** Understand the needs of real estate stakeholders. What features are most important for predicting house prices? How can these features be selected in a way that is both accurate and interpretable?

- **Ideate:** Brainstorm different methods for selecting features for house price prediction. This could involve using statistical methods, such as correlation analysis, or machine learning methods, such as recursive feature elimination.

- Feature Selection is the process of identifying and selecting the most relevant features from a dataset for a given machine learning task.
- The goal of feature selection is to improve the performance of the machine learning model by reducing the number of features and eliminating irrelevant or redundant features.

There are a variety of feature selection techniques. Some of the most common techniques include:

- **Correlation-based feature selection:** This technique selects features based on their correlation with the target variable. Features with high correlation with the target variable are more likely to be relevant for predicting the target variable, so they are selected.
- **Information gain-based feature selection:** This technique selects features based on their information gain. Information gain measures how much information a feature provides about the target variable. Features with high information gain are more likely to be relevant for predicting the target variable, so they are selected.
- **Recursive feature elimination (RFE):** This technique starts with all features and then recursively removes the least important feature until a desired number of features are remaining. The importance of a feature is measured using a variety of methods, such as cross-validation or the coefficient of determination (R-squared).
- **Model selection:** It is the process of choosing a suitable machine learning algorithm for a given machine learning task. The goal of model selection is to find an algorithm that is both accurate and efficient.

There are a variety of machine learning algorithms that can be used for house price prediction. Some of the most common algorithms include:

- **Linear regression:** Linear regression is a simple but effective algorithm for house price prediction. Linear regression models the relationship between the house price and the features using a linear function.
- **Random forest regressor:** Random forest regressor is a more complex algorithm that builds a multitude of decision trees to predict the house price. Random forest regressors are typically more accurate than linear regression models, but they can be more computationally expensive to train.
- **Gradient boosting regressor:** Gradient boosting regressor is another complex algorithm that builds a sequence of decision trees to predict the house price. Gradient boosting regressors are typically more accurate than random forest regressors, but they can be even more computationally expensive to train.

4. Model Selection:

- **Empathize:** Understand the needs of real estate stakeholders. How long is it acceptable for the model to train? How can the model be trained efficiently and effectively?
- **Ideate:** Brainstorm different ways to train the house price prediction model. This could involve using a variety of optimization techniques, such as stochastic gradient descent.
 - Choose machine learning algorithms suitable for regression tasks. Common models for predicting house prices include linear regression, decision trees, random forests, gradient boosting, and neural networks(e.g., deep learning models).
 - Experiment with multiple algorithms to determine which one provides the best performance for your specific dataset. You can also consider ensemble methods.

PYTHON PROGRAM:

```
# Import necessary libraries

from sklearn.feature_selection import SelectKBest, f_regression

from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, r2_score

import numpy as np

selector = SelectKBest(score_func=f_regression, k=k)

X_train_selected = selector.fit_transform(X_train, y_train)

# Model Selection

# Let's choose both Linear Regression and Random Forest Regressor
for comparison.

linear_reg_model = LinearRegression()

random_forest_model = RandomForestRegressor(n_estimators=100,
random_state=42)

# Train the models on the selected features

linear_reg_model.fit(X_train_selected, y_train)

random_forest_model.fit(X_train_selected, y_train)

# Evaluate the models on the test set

X_test_selected = selector.transform(X_test)

# Make predictions

linear_reg_predictions = linear_reg_model.predict(X_test_selected)
```

```
random_forest_predictions=random_forest_model.predict(X_test_selected)
```

```
# Evaluate model performance
```

```
def evaluate_model(predictions, model_name):
```

```
mse = mean_squared_error(y_test, predictions)
```

```
r2 = r2_score(y_test, predictions)
```

```
print(f"{model_name} Model Evaluation:")
```

```
print(f"Mean Squared Error (MSE): {mse}")
```

```
print(f"R-squared (R2) Score: {r2}\n")
```

```
# Performance Measure
```

```
elr_mse = mean_squared_error(y_test, pred)
```

```
elr_rmse = np.sqrt(lr_mse)
```

```
elr_r2 = r2_score(y_test, pred)
```

```
# Show Measures
```

```
result = ""
```

```
MSE : {}
```

```
RMSE : {}
```

```
R^2 : {}
```

```
".format(lr_mse, lr_rmse, lr_r2)
```

```
print(result)
```

```
# Model Comparison
```

```
names.append("elr")
```

```
mse.append(elf_mse)

rmse.append(elf_rmse)

r2.append(elf_r2)

evaluate_model(linear_reg_predictions, "Linear Regression")

evaluate_model(random_forest_predictions, "Random Forest
Regressor")
```

OUTPUT:

Linear Regression Model Evaluation:

Mean Squared Error (MSE): 10089009300.893988

R-squared (R2) Score: 0.9179971706834331

Random Forest Regressor Model Evaluation:

Mean Squared Error (MSE): 14463028828.265167

R-squared (R2) Score: 0.8824454166872736

MSE : 10141766848.330585

RMSE : 100706.33966305491

R^2 : 0.913302484308253

Evaluation:

- **Empathize:** Understand the needs of real estate stakeholders. What metrics are most important for evaluating the performance of the house price prediction model? How can the model be evaluated in a way that is both accurate and fair?
- **Ideate:** Brainstorm different metrics that could be used to evaluate the performance of the house price prediction model. This could

include mean absolute error (MAE), root mean squared error (RMSE), and R-squared.

- **Mean Absolute Error (MAE)**

MAE measures the average absolute difference between the predicted values and the actual target values. It provides insight into the average magnitude of errors made by the model.

- **Root Mean Squared Error (RMSE):**

RMSE is another common metric that calculates the square root of the average of squared differences between predicted and actual values. It provides information about the typical magnitude of errors and gives higher penalties to larger errors.

- **S-squared (R²):**

T-R-squared quantifies the proportion of the variance in the target variable that is explained by the model. It ranges from 0 to 1, where a higher value indicates a better fit. It is often used to assess how well the model captures the variation in the data.

Feature Engineering:

- Feature engineering is the process of creating new features or transforming existing ones to provide more meaningful information to the machine learning model. For the house price prediction project, we can consider the following feature engineering techniques:

1. Age of the House: Create a new feature that represents the age of each house by subtracting the year built from the current year. This feature can be informative as older houses may have different pricing dynamics compared to newer ones.

2. Square Footage per Bedroom: Calculate a new feature by dividing the total square footage of a property by the number of bedrooms. This metric can provide insights into the spaciousness of bedrooms, which can be a key factor in house pricing.

3. Bathrooms per Square Foot: Compute a new feature by dividing the number of bathrooms by the total square footage. This can capture the luxury level of bathrooms relative to the property's size.

4. School District Quality: Integrate external data from a third-party API to assess the quality of the school district in which each house is located. Properties situated in neighborhoods with better school districts often command higher prices.

Phase 1 Deliverables:

For Phase 1, the following deliverables are expected:

1. Clean and Preprocessed Dataset:

Provide a dataset where duplicates have been removed, missing values have been addressed, and categorical variables have been encoded. This dataset should be ready for use in machine learning models.

2. List of Engineered Features:

Document all the newly created features along with their descriptions and how they were calculated. This list should make it clear how each engineered feature adds value to the predictive model.

3. Data Preprocessing and Feature Engineering Report:

Create a detailed report that outlines the steps taken during data preprocessing and feature engineering. This report should cover the methods used, challenges encountered, and the rationale behind the decisions made at each step.

Next Steps

In Phase 2 of the project, we will proceed with the following tasks:

- **Model Selection and Training:** Select an appropriate machine learning algorithm(s) for house price prediction, train the chosen model(s) on the preprocessed data, and fine-tune hyperparameters if necessary.

- **Model Evaluation:** Evaluate the model's performance using various evaluation metrics on a held-out test dataset to assess its accuracy and generalization ability.
- **Additional Considerations**

In addition to the core data preprocessing and feature engineering steps, it's important to keep in mind the following considerations:

1. Machine Learning Algorithm Choice: The choice of machine learning algorithm(s) should be based on the dataset's characteristics and the desired performance metrics. Options include linear regression, random forests, gradient boosting, and neural networks.

2. Cross-Validation: Implement cross-validation to assess the model's performance without overfitting. Cross-validation involves splitting the data into folds, training on one subset, and evaluating on the others to obtain a more robust estimate of performance.

3. Interpreting Model Results: After training the model, focus on interpreting the results. Understand which features are most influential for predicting house prices, and explore how the model's predictions change as feature values vary.

Visualization and feature importance analysis can aid in interpretation.

CONCLUSION:

In Phase 1, we have established a clear understanding of our goal: to predict house prices using machine learning. We outlined a structured approach that includes data source selection, data preprocessing, feature selection, model selection, model training, and evaluation. This sets the stage for our project's successful execution in subsequent phases.