

House Price Prediction Project Documentation

Here's a structured documentation for your house price prediction project, including code explanations

Overview :-

This project aims to predict house prices using linear regression techniques, specifically Lasso and Ridge regression. The dataset contains features that influence house prices, and the model is trained to predict the price based on these features.

Requirements :-

- Python 3.x
- Pandas
- NumPy
- scikit-learn

Data Preparation :-

Dataset :- The dataset used is a CSV file containing house features such as:

- Neighborhood (Categorical: Rural, Suburb, Urban)
- Other relevant numerical features
- House Price (Target variable)

Data Loading and Preprocessing :-

```
```python
import pandas as pd

 • self.df = pd.read_csv(data)
 • self.df['Neighborhood'] = self.df['Neighborhood'].map({'Rural': 0, 'Suburb': 1,
 'Urban': 2}).astype(int)
```

**\*\*Explanation\*\*:** The `Neighborhood` feature is encoded into numerical values for model compatibility.

## Splitting the Data :-

```
```python

self.x_train, self.x_test, self.y_train, self.y_test = train_test_split(self.x, self.y, test_size=0.1,
random_state=42)
```

****Explanation**:** The dataset is split into training and testing sets, with 10% allocated for testing

Model Implementation

- **Lasso Regression** :-

```
```python
from sklearn.linear_model import Lasso
self.reg1 = Lasso(alpha=0.5)
self.reg1.fit(self.x_train, self.y_train)
```
```

****Explanation****: A Lasso regression model is created and trained on the training dataset.

- **Ridge Regression**

```
```python
from sklearn.linear_model import Ridge
self.reg2 = Ridge(alpha=0.5)
self.reg2.fit(self.x_train, self.y_train)
```
```

****Explanation****: A Ridge regression model is created and trained on the same training dataset.

Model Evaluation

Both models are evaluated using R^2 score and Mean Squared Error (MSE):

```
```python
from sklearn.metrics import mean_squared_error, r2_score
print("R2_score value =", r2_score(self.y_train, self.y_train_pred))
print("mean_squared_error =", mean_squared_error(self.y_train, self.y_train_pred))
```
```

Output Interpretation

- **** R^2 Score****: Indicates how well the model explains the variability of the target variable.
- ****Mean Squared Error****: Measures the average squared difference between predicted and actual values.

Error Handling

Error handling is implemented to catch exceptions during data loading and model fitting:

```
``python
```

```
except Exception as e:
```

```
    error_msg, error_line, error_type = sys.exc_info()
```

```
    print(f'error_line->{error_line.tb_lineno}, error_msg->{error_msg}, error_type->{error_type}')
```

Running the Program

To execute the program, run the following in your terminal:

```
python house_price_prediction.py
```

Ensure that the dataset path is correctly specified in the script.