BABCOCK UNIVERSITY
KNOWLEDGE TRUTH SERVICE

HOUSE PRICE PREDICTION- FULL PROJECT DOCUMENTATION

S/N	Name	Matric No
1.	BAMSON DAVID EYITOMILAYO	23/2732
2.	NWAOHA FAVOUR CHIJIOKE	23/1672
3.	ODIMEGWU PRECIOUS CHIJIOKE	23/1362
4.	MADUKAKU DIVINE CHEKWUBE	23/2837
5.	OLANREWAJU ADEBOYE EMMANUEL	23/2881
6.	ONWUBUALILI CHINEMEREM EMMANUEL	23/1551
7.	ORESOTU JOHN OLUWATOBI	23/1982
8.	OSUNNEYE BUSAYO ALEEM	23/1009
9.	OTOMIEWO EDWARD	23/2645
10.	OYEWUMI GERRARD DIEKOLOPEMI	23/0750

HOUSE PRICE PREDICTION – FULL PROJECT DOCUMENTATION

1.0 Introduction

This project focuses on building a machine learning application capable of predicting house prices based on various features such as demographics, housing characteristics, and geographic proximity to the ocean. The final deliverable is an interactive Streamlit web application that allows users to input housing attributes and receive an estimated price.

This documentation provides a clear overview of the workflow, preprocessing, model development, and deployment approach. Full source code is provided in the appendix.

2.0 Project Overview

The project uses several regression techniques to model the relationship between housing features and median house price. The implemented models include:

- Linear Regression
- LASSO Regression
- Ridge Regression

The dataset contains both numerical and categorical features, requiring appropriate preprocessing such as encoding, scaling, and polynomial feature expansion.

3.0 Source

The dataset used in this project was sourced from Kaggle, specifically from the "Housing Prices" notebook published by Mennatallah Nasr. The data includes variables such as median age, total rooms, total bedrooms, population, households, median income, and ocean proximity — all of which were used in preprocessing, feature transformation, and model training.

4.0 Data Preprocessing

4.1 Numerical Features

The numerical features used are:

- Median Age
- Total Rooms
- Total Bedrooms
- Population
- Households
- Median Income

Polynomial feature expansion was applied to capture more complex relationships within the data.

4.2 Categorical Features

The feature **ocean_proximity** was encoded using a fitted encoder. This encoding ensures consistency between training and inference.

4.3 Scaling

A StandardScaler fitted on the polynomial-transformed features ensures that all input variables contribute proportionally.
All preprocessing tools (encoder, scaler, and polynomial transformer) were serialized using pickle for later use in the Streamlit application.

5.0 Model Training

Three regression models were trained:

- **Linear Regression** - models basic linear relationships.
- **LASSO Regression** - performs feature selection and reduces overfitting by applying L1 regularization.
- **Ridge Regression** - addresses multicollinearity using L2 regularization.

All models were trained on the scaled polynomial feature dataset and saved as pickle files for later use in the web application.

6.0 Streamlit Application Workflow

The Streamlit app provides a simple, interactive interface where users enter housing details, select a preferred regression model, and instantly receive a predicted house price. The system automatically loads all preprocessing tools and models, processes the input, and generates an estimated house price.

7.0 Summary

This project demonstrates an end-to-end machine learning workflow: data preprocessing, model training, performance comparison, and deployment in an interactive web interface. Users can explore different regression models and understand how various features influence housing prices.

Appendix

Full source code for the streamlit application is as follows:

```
import streamlit as st
import pickle
import numpy as np

st.title("🏠 House Price Prediction App")

# Load Models
with open("regressor.pkl", "rb") as f:
    reg_model = pickle.load(f)

with open("lasso.pkl", "rb") as f:
    lasso_model = pickle.load(f)

with open("ridge.pkl", "rb") as f:
    ridge_model = pickle.load(f)


# Load Scaler & Encoder
with open("scaler.pkl", "rb") as f:
    scaler = pickle.load(f)

with open("encoder.pkl", "rb") as f:
    encoder = pickle.load(f)

with open("poly_feature.pkl", "rb") as f:
    poly = pickle.load(f)



# Model Selection
model_choice = st.selectbox(
    "Select Model",
    ["Linear Regression", "LASSO", "Ridge"]
)
```

```
# User Inputs
median_age = st.number_input("Median Age (Median age of the
house)", 0.0, 100.0, 20.0)
total_rooms = st.number_input("Total Rooms (Total number of rooms
within a block)", 0.0, value=1000.0)
total_bedrooms = st.number_input("Total Bedrooms (Total number of
bedrooms within a block)", 0.0, value=200.0)
population = st.number_input("Population (Total number of people
residing within a block)", 0.0, value=800.0)
households = st.number_input("Households (Total number of
households living in a block)", 0.0, value=300.0)
median_income = st.number_input("Median Income", 0.0, value=3.0)

ocean_proximity = st.selectbox(
    "Ocean Proximity (Location of the house w.r.t ocean/sea)",
    encoder.classes_
)


# Prediction Button
if st.button("Predict Price"):

    # Encode categorical variable
    ocean_encoded = encoder.transform([ocean_proximity])[0]

    # Prepare raw input
    input_data = np.array([[
        median_age,
        total_rooms,
        total_bedrooms,
        population,
        households,
        median_income,
        ocean_encoded
    ]])


    input_data= poly.transform(input_data)
    scaled_input = scaler.transform(input_data)
```

```
if model_choice == "Linear Regression":
    model = reg_model

elif model_choice == "LASSO":
    model = lasso_model

else:
    model = ridge_model

# Predict
prediction = model.predict(scaled_input)[0]

st.success(f"💰Predicted House Price: ${prediction:,.2f}")
```