Introduction:

In this document we see how to create the houseprice prediction model using python and let see how to develop the model step-by-step.

Step 1: Data Collection

Gather a dataset that includes information about houses. This dataset should contain attributes like square footage, number of bedrooms and bathrooms, location, and actual selling prices. You can find such datasets on real estate websites or platforms like Kaggle.

Step 2: Data Preprocessing

Clean the data by handling missing values, encoding categorical variables (like location), and scaling numerical features. This ensures that your data is in a format suitable for machine learning.

Step 3: Feature Selection

Choose the most relevant features that influence house prices. For instance, features like the number of bedrooms and the square footage are usually highly significant.

Step 4: Split the Data

Divide your dataset into two parts: a training set and a testing set. The training set is used to train the model, and the testing set is used to evaluate its performance.

Step 5: Model Selection

Select a machine learning model for regression. Linear regression is a simple and interpretable choice, but you can also explore more complex models like decision trees, random forests, or neural networks. Let's say we choose a neural network for this example.

Step 6: Model Architecture and Activation Functions

For the neural network model, you design its architecture. Common activation functions include:

- **Input Layer :** No activation function is applied to the input layer.

- **Hidden Layers :** You can use the Rectified Linear Unit (ReLU) activation function for hidden layers. It's defined as f(x) = max(0, x) and is effective at capturing non-linear relationships.

- **Output Layer** : Since we're predicting house prices (a continuous value), you can use a linear activation function (f(x) = x) for the output layer.

Step 7: Model Training

Train the neural network on the training data. This involves feeding the data through the network, calculating predictions, and adjusting the model's weights to minimize the prediction error. You can use optimization algorithms like Stochastic Gradient Descent (SGD) to update the weights.

Step 8: Model Evaluation

Assess the model's performance using the testing data. Common evaluation metrics for regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics measure how close the model's predictions are to the actual house prices.

Step 9: Hyperparameter Tuning

Fine-tune the model by adjusting hyperparameters, such as the learning rate, the number of hidden layers, and the number of neurons in each layer. Experiment with these parameters to achieve the best model performance.

Step 10: Model Deployment

Once satisfied with the model's performance, you can deploy it to make predictions. This could involve integrating the model into a web application or an API for real-time predictions.

Remember to document each step of your process, and provide detailed explanations in your 2-page document. Visualization of results and comparisons of different models or hyperparameters can be included to enhance the document's clarity. This will help others understand and replicate your work.