Phase 1: Problem Definition and Design Thinking

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem. Please think on a design and present in form of a document.

Problem Definition:

The problem is to predict house prices using machine learning techniques. The objective is to develop a model that accurately predicts the prices of houses based on a set of features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

Design Thinking:

Data Source: Choose a dataset containing information about houses, including features like location, square footage, bedrooms, bathrooms, and price.

Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

Feature Selection: Select the most relevant features for predicting house prices.

Model Selection: Choose a suitable regression algorithm (e.g., Linear Regression, Random Forest Regressor) for predicting house prices.

Model Training: Train the selected model using the preprocessed data.

Evaluation: Evaluate the model’s performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.

Step 1: Download and Install Git

1. Visit the official Git website: <https://git-scm.com/>
2. Download the appropriate version of Git for your operating system (Windows, macOS, or Linux).
3. Run the installer and follow the on-screen instructions to complete the installation.
4. Open a terminal or command prompt and verify the installation by typing git—version .

Step 2: Download and Install Visual Studio Code

1. Go to the Visual Studio Code website: <https://code.visualstudio.com/>
2. Download the installer for your operating system (Windows, macOS, or Linux).
3. Run the installer and follow the installation prompts.
4. Launch Visual Studio Code.

Step 3: Create a GitHub Account

1. Open a web browser and go to <https://github.com/>
2. Click on the “Sign up” button.
3. Follow the registration process, providing your username, email address, and password.
4. Complete the verification process if prompted.

Step 4: Create a GitHub Repository

1. Log in to your GitHub account.
2. Click on your profile icon in the upper right corner and select “Your repositories” from the dropdown menu.
3. On the “Repositories” page, click the green “New” button.
4. Fill in the required information for your new repository, including the repository name, description, visibility, and other settings.
5. Optionally, you can choose to initialize the repository with a README file or add a .gitignore file for your specific project.
6. Click the green “Create repository” button to create your GitHub repository.

Step 5: Create a Local Folder

1. Minimize any open windows on your computer to see your desktop.
2. Right-click on an empty area of your desktop.
3. Hover over “New” in the context menu.
4. Click on “Folder” to create a new folder.
5. Give your folder a meaningful name, like “MyProject.”

Step 6: Open the Folder in Visual Studio Code

1. Launch Visual Studio Code.
2. Click on “File” in the top-left corner.
3. Select “Open Folder” from the dropdown menu.
4. Browse to your desktop and select the folder you created in Step 5 (e.g., “MyProject”).
5. Click the “Open” button to open the folder in Visual Studio Code.

Step 7: Clone Your GitHub Repository

1. In Visual Studio Code, open the integrated terminal by clicking on “View” in the top menu and selecting “Terminal” or using the keyboard shortcut (Ctrl+ on Windows/Linux or Cmd+ on macOS).
2. Use the git clone command to clone your GitHub repository by pasting the HTTPS URL of your repository. Replace repository\_url with the actual URL.

Git clone <repository\_url>

1. Navigate to the newly created repository folder using the cd command:

Cd <repository\_name>

Step 8: Check Git Status

1. To check the status of your local repository, enter the following command:

Git status

Step 9: Modify the README File

1. Open the README file in your repository folder using Visual Studio Code.
2. Make the desired modifications to the README file.

Step 10: Check Git Status Again

1. Return to the terminal in Visual Studio Code.
2. Use the gitstatus command again to see the changes you made:

Git status

Step 11: Add Modifications to Staging Area

1. To stage your changes for a commit, use the gitadd command:

Git add README.md

Step 12: Commit Your Changes

1. Commit your staged changes with a descriptive message:

Git commit -m “Updated README file”

Step 13: Push Changes to GitHub

1. Push your committed changes to your GitHub repository:

Git push

Step 14: Create a New Branch

1. To create a new branch, use the gitbranch command followed by the desired branch name:

Git branch branch\_name

Step 15: Switch to the New Branch

1. To switch to the newly created branch, use the gitcheckout command:

Git checkout branch\_name

Step 16: Check Your Current Branch

1. To confirm the branch you’re currently working on, use the gitbranch command:

Git branch

You’ve now completed the entire process of setting up a development environment, downloading and installing Git and Visual Studio

NOTE: File Naming Convention: AI\_Phase1

After completion upload your file to your private GitHub account. Please give access to your faculty evaluators of your college and industry evaluator [ [IndustryEvaluator@skillup.online](mailto:IndustryEvaluator@skillup.online) ] to your private GitHub repository for evaluation process Go to the Project Submission Part 1 section and add your college code, the link of your GitHub in the space provided, upload your documents, and click on submit.

Phase 2,

Design Thinking:

• Data Source: Choose a dataset containing information about houses, including features like

Location, square footage, bedrooms, bathrooms, and price.

• Data Preprocessing: Clean and preprocess the data, handle missing values, and convert

Categorical features into numerical representations.

• Feature Selection: Select the most relevant features for predicting house prices.

• Model Selection: Choose a suitable regression algorithm (e.g., Linear Regression, Random Forest

Regressor) for predicting house prices.

• Model Training: Train the selected model using the preprocessed data.

• Evaluation: Evaluate the model’s performance using metrics like Mean Absolute Error (MAE),

Root Mean Squared Error (RMSE), and R-square

AI\_phase4 Predicting house prices using machine learning involves several phases: data preprocessing, feature selection, model training, and evaluation. Here’s a brief overview of each phase, and I’ll provide Python code for feature selection as an example:

\*\*Phase 1: Data Preprocessing\*\*

* + Data cleaning and handling missing values.
  + Encoding categorical variables.
  + Scaling or normalizing numerical features.
  + Splitting the data into training and testing sets.

\*\*Phase 2: Feature Selection\*\*

Feature selection is essential to choose the most relevant features for your model. Here’s an example using Python’s `scikit-learn` library with a hypothetical dataset:

```python

From sklearn.feature\_selection import SelectKBest

From sklearn.feature\_selection import f\_regression

# Assuming X contains your feature data and y contains target prices

X\_new = SelectKBest(score\_func=f\_regression, k=5).fit\_transform(X, y)

```

This code uses the F-regression method to select the top 5 features based on their relevance to predicting house prices.

\*\*Phase 3: Model Training\*\*

You can choose from various regression algorithms, such as Linear Regression, Random Forest, or Gradient Boosting. Here’s an example of training a simple Linear Regression model:

```python

From sklearn.linear\_model import LinearRegression

Model = LinearRegression()

Model.fit(X\_new, y)

```

\*\*Phase 4: Evaluation\*\*

You should evaluate the model’s performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared. Here's an example:

```python

From sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

Import numpy as np

Predictions = model.predict(X\_new)

Mae = mean\_absolute\_error(y, predictions)

Mse = mean\_squared\_error(y, predictions)

R2 = r2\_score(y, predictions)

Print(f”Mean Absolute Error: {mae}”)

Print(f”Mean Squared Error: {mse}”)

Print(f”R-squared: {r2}”)

``