AI ASSIGNMENT - 2

Step 1: Import the necessary libraries

python

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

Step 2: Load the dataset

python

# Assuming the dataset is stored in a CSV file named 'house\_data.csv'

df = pd.read\_csv('house\_data.csv')

Step 3: Preprocess the dataset

python

# Drop any irrelevant columns if needed

df = df.drop(['Date'], axis=1)

# Handle missing values

df.fillna(df.mean(), inplace=True)

# One-hot encode categorical variables (if any)

df\_encoded = pd.get\_dummies(df, drop\_first=True)

# Split the dataset into features (X) and target variable (y)

X = df\_encoded.drop('Price', axis=1)

y = df\_encoded['Price']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the numerical features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

Step 4: Build the ANN model

python

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)))

model.add(Dense(64, activation='relu'))

model.add(Dense(1)) # Output layer

model.compile(optimizer='adam', loss='mean\_squared\_error')

Step 5: Train the model

python

model.fit(X\_train\_scaled, y\_train, epochs=10, batch\_size=32, verbose=1)

Step 6: Test the model

python

y\_pred = model.predict(X\_test\_scaled)

# Evaluate the model

loss = model.evaluate(X\_test\_scaled, y\_test)