PHASE 2 ASSIGNMENT

PROJECT TITLE: INNOVATION

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.

GITHUB LINK:

https://github.com/Prasikasaravanan/predicting-house-price-using-machine-learning.git

DOCUMENT:

Predicting House Prices Using Machine Learning Innovation on Ensemble method and deep learning for Dataset

DATASET LINK ON: Predicting House Prices

https://www.kaggle.com/datasets/vedavyasv/usa-housing

Designing a process to apply ensemble methods and deep learning techniques to a dataset involves several steps. Here's a general guide that you can follow:

1. Understand the Dataset:

Explore and analyze the dataset to gain insights into its structure, patterns, and characteristics.

Identify the type of problem (classification, regression, etc.) and the nature of the data (structured, unstructured).

2. Data Preprocessing:

Handle missing values, if any.

Encode categorical variables using techniques like one-hot encoding.

Standardize or normalize numerical features.

Split the dataset into training and testing sets.

3. Ensemble Methods:

a. Bagging (Bootstrap Aggregating):

Choose a base model (e.g., Decision Trees).

Create multiple subsets of the training data through bootstrap sampling.

Train a model on each subset.

Aggregate predictions through averaging (for regression) or voting (for classification).

b. Boosting:

Choose a weak learner (e.g., Decision Trees).

Train a weak model on the entire dataset.

Adjust the weights of misclassified instances and train the next model.

Repeat until a predefined number of models are built.

c. Random Forest:

Choose the number of trees and other hyperparameters.

Train multiple decision trees using random subsets of features.

Aggregate predictions through voting.

4. Deep Learning:

a. Neural Network Architecture:

Design the architecture of your neural network, considering the problem type.

Decide on the number of layers, nodes, and activation functions.

b. Model Compilation:

Choose an optimizer, loss function, and evaluation metric.

Compile the model.

c. Model Training:

Train the model on the training data.

Adjust hyperparameters (learning rate, batch size) based on performance.

5. Ensemble of Deep Learning Models:

Train multiple neural networks with different initializations or architectures.

Combine predictions using methods like averaging or voting.

6. Evaluation:

Evaluate the performance of individual models, ensemble methods, and deep learning models on the validation set.

Fine-tune hyperparameters based on validation performance.

7. Testing:

Evaluate the final models on the test set to assess generalization performance.

8. Hyperparameter Tuning:

Use techniques like grid search or random search to find optimal hyperparameters for both ensemble methods and deep learning models.

9. Feature Importance (Optional):

For ensemble methods, analyze feature importance to understand which features contribute the most to predictions.

10. Documentation:

Keep a record of the models, hyperparameters, and evaluation metrics for future reference.

11. Iterate and Refine:

Based on performance, iterate on the models, adjust hyperparameters, and refine the process.

12. Deployment (if applicable):

Deploy the final model in a production environment if the performance meets the desired criteria.

Remember that the specific steps might vary depending on the nature of your dataset and the problem you are trying to solve. Adjustments and iterations are often necessary for optimal results.

SUBMITTED BY,

STUDENT REG NO: 711221104039

NAAN MUDHALVAN: au711221104039