HW4 for Faculty Session: (Used ChatGPT for few questions, however, all writing is from my understanding)

Part 1:

1. Garbage In, Garbage Out (GIGO) (a) GIGO's implications for machine learning performance (5 points):

"Garbage In, Garbage Out" (GIGO) refers to the fact that a machine learning model will generate inaccurate or deceptive results if the input data is of low quality, such as incorrect, inconsistent, missing, or irrelevant. Inaccurate data cannot be compensated for by even the most advanced algorithms. To put it briefly, the quality of the input directly affects the quality of the output.

(b)

Missing Values: Null or empty entries can be found in a lot of datasets. Improper handling of this may result in training errors or lower model accuracy.

Data may have categorical or inconsistent formats, such as "yes/No," "Y/N," or various date formats. Inconsistency can result in misinterpretation or processing failure because models require consistent, numerical input.

Noisy data or outliers: Particularly for algorithms like linear regression, extreme values have the potential to distort the model's learning. Additionally, data noise can confuse the learning process and lower the signal-to-noise ratio.

2. (Used AI help)

Fill in the missing values (imputation):

- *Example:* Use the average (mean) to fill missing numbers.
- *Good for:* When only a few values are missing.
- *Problem:* It might not be accurate and can reduce variation in the data.

Remove rows or columns with missing values:

- *Example:* If a row has too many blanks, just delete it.
- *Good for:* When the dataset is big and the missing part is not important.
- *Problem:* You lose data, which might affect the model if too much is removed.

3.

When features (columns) have widely disparate ranges, some models become confused. For instance, the model may be unfairly impacted more by the first column if it contains values from 1 to 1000 and another from 0 to 1.

K-Nearest Neighbors (KNN) is a model that requires scaling because it makes use of distances between data points.

A model that doesn't require scaling: Decision trees don't care about scale because they only consider splitting points and not distances.

Part 2:

4)

To ensure that the model learns well, tunes appropriately, and generalizes to new data, the dataset should be split.

The portion of the data that the model learns from is called the training set. It is employed to modify internal weights and identify patterns.

Model parameters (such as the number of layers or learning rate) are adjusted using the validation set. Without affecting the test set, it assists in determining which model settings are most effective.

Test Set: This is used just once, following training and fine-tuning, to assess the model's performance on unknown data. It provides a last, frank assessment of the model's practicality.

5)

a) Overfitting occurs when a model learns the training data too thoroughly, including non-generalizable details and noise. Accuracy in training increases significantly. As a result of the model's poor performance on fresh, untested data, test accuracy decreases.

(b) The test set serves as "fresh eyes" for the model, demonstrating its performance on previously unseen data. The model is overfitting if performance is excellent on training data but subpar on test data. It would be challenging to identify this without a distinct test set.

6)

A loss function calculates the deviation between the actual values and the model's predictions. It functions similarly to a "score," indicating how poorly the model is performing. The model improves its predictions during training in an effort to reduce the loss. Better predictions result from a lower loss.

For instance, Mean Squared Error (MSE) is a frequently used loss function in regression, where the model seeks to minimize the average squared discrepancies between expected and actual values.

7)

The process of developing new input features from preexisting data in order to enhance the model's learning capabilities is known as feature engineering.

Why it's important Better features aid in the model's better understanding of the data, which frequently results in improved model performance. As an illustration, consider a dataset that contains a person's birthdate.

By deducting the birth year from the current year, you can create a new feature called "age." The complete date of birth may not be as helpful to the model as this new feature.

Part 3:

8. The entire dataset might not be accurately represented by a single validation set, particularly if it is small or dispersed unevenly. This can result in a high variance in the model's performance, which means that the split's random chance may determine how good or bad the model appears rather than its true capabilities.

9a) K-Fold The data is divided into K equal parts (folds) by cross-validation. Using a different fold as the validation set and the remaining folds as training, the model is trained K times.

This lowers variance and provides a more accurate estimate of model performance by ensuring that each data point appears in the validation set once.

9b) Five training runs of the model, one for each fold.

10.

Testing your trained model on a brand-new dataset that wasn't used at all for training, validation, or tuning is known as external validation.

Because it demonstrates how well your model generalizes to actually unseen data from a different source or time, it is more robust and more representative of real-world use. This aids in identifying bias or overfitting that internal validation (such as cross-validation) might overlook.

11. Overly optimistic performance results can result from data leakage, which happens when information from outside the training dataset is inadvertently used to build the model. If preprocessing steps like scaling, encoding, or imputing are used before data is divided into training and testing sets, this usually occurs during those steps.

As a result, the model "sees" information that it shouldn't have had access to during training, which makes it function well in tests but not in practical situations. To guarantee that model evaluation is equitable and accurately reflects generalization, data leakage must be prevented.

Part 4: (Used some ChatGPT for insights)

12) Making a trained machine learning model usable so that it can be applied to real-world data and produce predictions is the primary objective of model deployment. Deployment fills the gap between training the model and putting it to use in production, which could be done through a system, app, or website.

13) You can reuse a trained model later without having to retrain it from scratch by saving it (for example, by using pickle). In order to enable others (or systems) to load and run the model effectively during deployment or sharing, this step is crucial because it maintains the learned parameters and structure. It saves time and guarantees consistency, particularly in cases where training is costly or time-consuming.

14) Batch Prediction Scenario: Let's say a business wishes to generate weekly reports to forecast which clients are most likely to click on advertisements the following week. There is no need for instant results because they can process thousands of records at once using batch predictions, either overnight or on a schedule.

Real-time Prediction Scenario: The system must instantly determine which advertisement to display based on a user's behavior when they visit a website. Real-time predictions made through an API are perfect in this situation because they can respond quickly—within milliseconds.

15) When code functions properly on a developer's computer but malfunctions on another system because of environmental differences (such as Python versions, libraries, or OS settings), this is known as the "Works on My Machine" issue.

By encapsulating the application, its dependencies, and its environment in a container that can operate consistently anywhere, Docker addresses this issue. Even if you're not coding it directly, this guarantees consistency and minimizes deployment problems.