Patan College for Professional Studies

Kupondole, Lalitpur

Week 8: Machine Learning Basics + Supervised Learning: Linear Models

Week 9: Advanced Supervised Learning

INDIVIDUAL ASSIGNMENT

Name: Prabesh Kattel

Level: L5

Student ID: 2402412077

Subject: Data Science with AI

Section A: Theory Questions

1. **Explain the typical steps involved in a machine learning pipeline.**

- The typical steps involved in a machine learning pipeline are as follows:
  - **a) Data Collection:** This step includes the collection of relevant data from the various sources.
  - **b) Data Preprocessing:** Data preprocessing is the process of cleaning the data, i.e., handling missing data, outliers, performing normalization, and encoding categorical variables.
  - **c) Feature Engineering:** After the data is cleaned and normalized into their standard forms, the Data is selected and transformed to create meaningful features. While selecting irrelevant features is removed, those with low variance and high correlation are removed.
  - **d) Model selection:** The main purpose of model selection is to choose the best algorithm for the problem. Here, the types of problems and data size with its complexity of the system are analyzed. Small datasets are included in simple models, so we use Logistic Regression, whereas large datasets are included in deep learning, so we use neural networks.
  - e) **Training the Model:** The main purpose of training the data is to fit the model to the training data. Splitting data, Model fitting, and handling overfitting are the steps while training the model.
  - **f) Model Evaluation:** The Model is evaluated to know how well the model generalizes. Regression, Classification, and cross-validation is calculated and examined.
  - **g) Hyper parameter Tuning:** Optimizing the model parameters for better performance is called hyper parameter tuning.
  - h) **Deployment:** Deployment is the process of integrating the model into performance for real-world predictions. E.g. hosting the Chabot in AWS Lambda.

2. **Differentiate between supervised and unsupervised learning with real-world examples.**

Supervised Learning	**Unsupervised Learning**
It includes features and targets.	It includes only the features.
The main aim of this learning is to predict the outcomes (classification or regression).	The main purpose of this learning is finding the patterns or structure (clustering or dimension reduction).
Example: predicting the house prices (regression), spam detection (classification)	Example: Customer Segmentation (clustering)

3. **What is the goal of linear regression? Explain the concept of the line of best fit.**

- Linear regression is a supervised learning algorithm used to predict a continuous output based on one or more input features.
- The goal is to find the best-fitting straight line that explains the relationship between the independent variable(s) and the dependent variable. This line helps in making predictions for unseen data.
- Line of Best Fit:

- It is a straight line that best represents the data on a scatter plot which is also called as regression line. It minimizes the sum of squared differences (errors) between actual data points and predicted values (Least Squares Method).
- The equation of the line is: $y=mx+by = mx + by=mx+b$
- Here, x=input variable, y=predicted variable, m=slope of line, b=intercept on y-axis

Purpose of the Line of Best Fit:

- To capture the trend or pattern in the data.
- To make future predictions based on the given input.
- To analyze relationships between variables (e.g., how one variable increases/decreases in relation to another).

The example is: Predicting house prices:

x = house size in square feet, y = price of the house

The line of best fit shows how house price typically increases with size.

4. How is the R^2 score interpreted? What does an R^2 of 0.85 indicate?

- R-squared score is a statistical measure that shows how well the regression line fits the data**.** It represents the proportion of the variance in the dependent variable that is explained by the independent variable(s)**.** The value of R^2 ranges from 0 to 1**:**
    - **$R^2 = 1$** → Perfect fit (100% of the variance is explained)
    - **$R^2 = 0$** → No fit (model explains none of the variance)

- Interpretation of R^2:

- Higher R^2 means a better fit between the model and the actual data.
- It helps in evaluating the accuracy of a regression model.

- $R^2 = 0.85$ means:

→ 85% of the variability in the dependent variable can be explained by the model.

→Only 15% of the variation is due to other unknown or random factors.

→This indicates a strong relationship between the input and output variables.

→The model is considered to be a good predictor in this case.

5. Explain how logistic regression differs from linear regression.

- Logistic regression differs from linear regression due to the following points:

S.N.	Logistic regression	Linear Regression
1	Used for classification tasks i.e. yes/no, accept/decline	Used for predicting the continuous values i.e. price, age, weight and height.
2	The output is the probability between 0 and 1.	The output would be the real number.
3	Sigmoid function is used.	Linear function is used.

4	Predicting whether the assignment is copied or not.	Used in predicting house prices based on the features.

6. What is the sigmoid function, and how does it help in classification?

- The sigmoid function is a mathematical function that converts any real-valued number into a value between 0 and 1**.** It is commonly used in binary classification problems (like predicting 0 or 1). The formula to calculate the sigmoid function is $\sigma(z) = \frac{1}{1+e^{-z}}$, where z is the weighted sum of units. In addition to these characteristics of the graph, is mentioned:

- S- S-shaped curved, also known as the logistic curve.
- Output seems to be 0 as z becomes very negative.
- Output seems 1 as z becomes very positive.
- Output is 0.5 when z=0.

It helps in classification by converting the linear output into a probability score between o and 1. In logistic regression, output greater than 0.5 is classified as Class 1, whereas output less or equal to 0.5 is classified as Class 0. In addition to this, it allows model to make probabilistic decisions instead of just yes/no outputs.

7. Define Accuracy, Precision, Recall, and F1-score with suitable use-case examples.

Terms	**Definition**	**Formula**	**Use-Case Example**
Accuracy	The ratio of correctly predicted observations to the total observations.	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	In a handwriting recognition system, if 95 out of 100 digits are correctly classified, accuracy = 95%.
Precision	The ratio of correctly predicted positive observations to the total predicted positive observations.	$Precision = \frac{TP}{TP + FP}$	In a spam filter, if 10 emails are marked as spam and 8 are actually spam, precision = 80%. High precision means few false alarms**.**
Recall	The ratio of correctly predicted positive observations to all actual positives.	$Recall = \frac{TP}{TP + FN}$	In a cancer detection test, if 90 out of 100 actual cancer cases are correctly detected, recall = 90%. High recall means few missed detections**.**
F1-score	The harmonic mean of Precision and Recall. It balances both metrics.	$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$	In fraud detection, where both false positives and false negatives are costly, the F1-score is a better measure than just accuracy.

8. How does KNN work for classification problems?

- K-Nearest Neighbors (KNN) is a supervised learning algorithm used for classification and regression**.** In classification, it predicts the class of a data point based on the majority class among its 'K' nearest neighbors**.** Below, I have mentioned the working mechanism for classification problems:
    - Choosing value of K (considering the number of neighbors)
    - Calculating the distance between the new data point (testing) and all training data points.
    - Identifying the K nearest neighbor's i.e. smallest distance.
    - Vote for the majority class among the K neighbors.
    - Assign the class with the highest votes to the test data point.

9. Explain the impact of different distance metrics (Euclidean vs Manhattan) on KNN performance.

- In K-Nearest Neighbors (KNN)**,** the choice of distance metric plays a key role in how the algorithm measures similarity between data points. Two commonly used metrics are Euclidean distance and Manhattan distance**.** Euclidean distance calculates the shortest straight-line distance between two points, while Manhattan distance measures the distance by only moving along horizontal and vertical paths — like navigating a grid of city blocks. The chosen metric affects which neighbors are considered "closest" during classification or prediction.
- The impact on KNN performance depends on the structure and distribution of the data. Euclidean distance works well when features are continuous and the data is dense and evenly distributed. However, it can be sensitive to large differences in feature values. Manhattan distance, on the other hand, is often more robust when features are high-dimensional or when data has many outliers, as it doesn't exaggerate the effect of large differences in a single dimension. Choosing the right distance metric can significantly improve accuracy, especially in complex or noisy datasets.

10. Why is choosing the right value of K important? What happens if K is too small or too large?

- In K-Nearest Neighbors (KNN)**,** choosing the right value of K (number of neighbors) is crucial because it directly affects the model's accuracy and ability to generalize. The value of K determines how many neighboring data points will be considered when making a prediction. A well-chosen K balances between overfitting and under fitting, helping the model perform well on both training and unseen data.
- If K is too small (e.g., K = 1), the model becomes too sensitive to noise or outliers**,** leading to overfitting — it may perform very well on training data but poorly on test data. On the other hand, if K is too large**,** the model may include points from different classes in its prediction, causing under fitting — it may miss important patterns and oversimplify the decision boundary. Therefore, selecting an appropriate K is essential for maintaining the model's balance and accuracy.

11. Explain the concept of decision boundary and margin in SVM.

- In Support Vector Machine (SVM), the decision boundary is a hyperplane that separates data points of different classes. In two-dimensional space, this boundary appears as a straight line, while in higher dimensions, it's a flat surface (hyperplane). The goal of SVM is to find the optimal decision boundary that not only separates the classes but does so in a way that provides the best possible separation. This boundary is determined using only the most critical data points from each class, known as support vectors.

- The margin in SVM refers to the distance between the decision boundary and the closest data points from each class (the support vectors). SVM aims to maximize this margin, as a larger margin typically leads to better generalization on unseen data and reduces the risk of overfitting. In essence, SVM chooses the boundary that is farthest away from the nearest data points of both classes, ensuring the model makes confident and reliable predictions.

12. What is the kernel trick, and why is it useful?

- The kernel trick is a technique used in machine learning (especially in Support Vector Machines**)** to transform data into a higher-dimensional space without explicitly computing the transformation. It allows algorithms to learn complex patterns using simple linear models by operating in a transformed feature space.
- It is useful because some datasets are not linearly separable in their original form. By mapping data to a higher dimension, the algorithm can find a hyperplane (line or surface) that separates the classes.

13. Compare SVM and KNN in terms of training time, performance, and scalability.

Terms	**Support Vector Machine (SVM)**	**K-Nearest Neighbors**
Training Time	- High training time (especially with large datasets) - Requires optimization to find the best decision boundary	- Very low training time - Just stores the data (lazy learner
Performance	- High accuracy on high-dimensional and complex data - Good with clear margins	- Performs well on small to medium datasets - Sensitive to noise and irrelevant features
Scalability	- Scales well with dimensionality - Slower on very large datasets	- Does not scale well - Becomes very slow as dataset size grows

14. What is Bayesian optimization? Give a real-life example.

- Bayesian Optimization is a sequential model-based optimization technique used to find the best parameters for a function that is expensive or time-consuming to evaluate. It builds a probabilistic model (usually using Gaussian Processes) of the function and uses it to choose the most promising next point to evaluate.
- **Example : Tuning the temperature and baking time for a perfect cake**
- You're baking cakes to find the best oven temperature and baking time.
- But each attempt takes 1 hour and costs ingredients, so you can't test every combination.
- Bayesian Optimization:
    - Starts with a few test runs (e.g., 170°C for 30 min, 180°C for 25 min).
    - Builds a model of how temperature and time affect cake quality.
    - Suggests the most promising settings to try next.
    - Gradually finds the best combination with fewer attempts.

15. Describe the process of hyper parameter tuning with examples.

- Hyper parameter tuning is the process of finding the best combination of settings (called hyperparameters) that optimize a machine learning model's performance. Unlike model

parameters that are learned during training (like weights in linear regression), hyper parameters are set before training begins — such as learning rate, number of neighbors (K in KNN), maximum depth in decision trees, or regularization strength in SVM. Tuning these values is crucial because the right combination can significantly improve the model's accuracy, speed, and generalization on unseen data.

- There are several common methods for hyper parameter tuning. In Grid Search, the algorithm tests all possible combinations from a defined list of hyper parameter values. In Random Search**,** combinations are selected randomly, often faster for large spaces. A more advanced method is Bayesian Optimization, which intelligently selects the next set of hyper parameters to try based on past performance. For example, in a decision tree classifier, tuning max_depth, min_samples_split, and criterion can help prevent overfitting and boost test accuracy. Tools like scikit-learn's GridSearchCV automate this process using cross-validation to ensure robust evaluation.