Machine Learning

Q1)a. Explain Lasso Regression. Explain how Lasso Regression is used for

feature selection.

Ans: **Lasso Regression** (Least Absolute Shrinkage and Selection Operator) is a type of linear regression that uses **L1 regularization** to penalize the absolute size of the coefficients. The regularization term is added to the cost function to reduce the model's complexity and prevent overfitting.

Where:

1. **RSS** is the residual sum of squares.
2. **λ** is the regularization parameter (controls the strength of the penalty).
3. **βi** are the model coefficients.

Feature Selection in Lasso:

Lasso performs automatic feature selection by forcing the coefficients of some features to be exactly zero. The process works as follows:

* The L1 penalty added to the cost function (the sum of absolute values of coefficients) encourages sparsity.

* Features with coefficients shrinking to zero are effectively removed from the model.

* This allows Lasso to select only the most important feature & naking the model more efficient and reducing multicollinearity.

In short, Lasso regression not only fits the data but also helps identify and retain only the most relevant features for the prediction

task.

Q1)b. Define different regression models.

Ans: Here's a concise overview of different regression models:

1. Linear Regression: Models the relationship between a dependent variable and independent variables using a straight line. Best for linear relationships. 2. Ridge Regression: A variant of linear regression with L2 regularization to prevent overfitting by shrinking coefficients.

3. Lasso Regression: Similar to Ridge but uses L1 regularization, which can shrink some coefficients zero, thus performing feature selection.

4. Elastic Net Regression: Combines L1 and L2 regularization, balancing feature selection and coefficient shrinkage.

5. Polynomial Regression: Extends linear regression by modeling nonlinear relationships using polynomial terms.

6. Logistic Regression: Used for binary classification, mo-'-Is the probability of a categorical outcome (0 or 1)

7. K-Nearest Neighbors Regression

(KNN): Predicts the target value based on the average of the k nearest neighbors, useful for non-linear relationships.

Each model is suited for different types of data and tasks.

Q1)c. Describe the bias-variance trade-off and its relationship to under fitting and overfitting.

Ans: The bias-variance trade-off is a fundamental concept in machine learning that describes the balance between two sources of error in a

model:

1. Bias: The error introduced by assuming a simplistic model that doesn't capture the underlying patterns of the data. High bias leads to underfitting, where the model is too simple and performs poorly on both training and test data.

2. Variance: The error due to the model's sensitivity to fluctuations in the training data. High variance leads to overfitting, where the model comes too complex and captures noise or irrelevant patterns

Relationship to Underfitting and Overfitting:

* Underfitting occurs when the model has high bias (too simple), resulting in poor performance on both training and test

data.

* Overfitting occurs when the model has high variance (too complex), resulting in excellent performance on training data but poor generalization to new, unseen data.

The trade-off is about finding the right balance: minimizing both bias and variance.

Q3)a. Explain kernel methods which are suitabie for SVM.

Ans: Kernel methods in Support Vector Machines

(SVM) are used to transform non-linearly separable data into a higher-dimensional space where it becomes linearly separable.

This is achieved without explicitly computing the coordinates of the data in this higher-dimensional space, using a kernel function to compute the inner product in the transformed space directly.

Common Kernel Functions:

1. Linear Kernel: No transformation is applied, and the data is assumed to be already linearly separable.

* K(x, y) = xy

2. Polynomial Kernel: Maps the data into a higher-dimensional polynomial feature space.

* K(x, y) = (xy + c) d

where C is a constant and d is the degree of the polynomial.

3. Radial Basis Function (RBF) Kernel:

Maps the data into an infinite dimensional space and is effective

Role in SVM:

* Kernel methods allow SVM to learn non-

linear decision boundaries by implicitly working in a higher-dimensional space, without the need to compute the

transformed data explicitly.

* This helps SVM classify complex data that is not linearly separable in the original feature space.

Q3)b. What are advantages and disadvantages of K-NN?

Ans: Advantages of K-Nearest Neighbors (K-NN):

1. Simple and Intuitive: Easy to understand and implement.

2. No Training Phase: K-NN is a lazy learner, meaning it doesn't require a training phase, which can be beneficial for large datasets.

3. Non-Parametric: It makes no assumptions about the data distribution, useful for complex and non-linear data.

4. Flexible: Can be used for both classification and Regression tasks.

Disadvantages of K-NN:

1. Computationally Expensive: The prediction phase is slow, as it requires calculating distances to all training samples.

2. Sensitive to Irrelevant Features:

Performance can degrade with high-dimensional or noisy data.

3. Memory Intensive Requires storing the entire training dataset for prediction which can be inefficient for large

datasets.

4.Choosing K: The choice of the right value for K can significantly affect performance, and it might require cross-validation to tune.

Q3)c. What are different distance metrics are used in K-NN?

Ans: In K-Nearest Neighbors (K-NN), different distance metrics are used to calculate the

similarity between data points. Common ones include:

1. Euclidean Distance:

* The most widely used metric, measures the straight-line distance

between two points.

$$\textbf{Euclidean Distance} = |X - Y| = \sqrt{\sum_{i=1}^{i=n} (x_i - y_i)^2}$$

X: Array or vector X

Y: Array or vector Y

x_i: Values of horizontal axis in the coordinate plane

y_i: Values of vertical axis in the coordinate plane

n: Number of observations

2. Manhattan Distance:

Measures the sum of the absolute differences of the coordinates, useful in grid-like data.

* Formula: d(x, y) = Li, |Xi - Yil

3. Minkowski Distance:

* Generalizes Euclidean and Manhattan distances by introducing a parameter p.

Formula: $$D = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{1/p}$$

4. Chebyshev Distance:

Measures the maximum absolute difference along any coordinate dimension.

* Formula: d(x, y) = maxi Xi - Y|

Each distance metric is suitable for different types of data and use cases, depending on the problem at hand.

Q6)a. What is isolation model?

Ans: The Isolation Forest model is an anomaly section algorithm that works by isolating through randomly selected splits. The core idea is that anomalies liners are easier to isolate than normal i points, which tend to be more clustered

together.

How It Works:

1. Isolation: The algorithm builds multiple random trees by recursively selecting random features and splitting the data at random values.

2. Isolation of Anomalies:Anomalies are isolated in fewer steps because they are

different from the majority of the data.

3. Scoring: The number of steps required to isolate a data point is used as a measure of its anomaly score — fewer steps indicate an anomaly.

Key Advantages:

1. Efficient: It is faster and more scalable

compared to traditional methods, like clustering or distance-based methods.

2. No Assumptions : does not require assumptions about the distribution .

Q6)b. Explain hierarchical clustering with an example.

Ans : Hierarchical Clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. It can be divided into two types:

1. Agglomerative (Bottom-Up): Starts with each data point as a separate cluster and iteratively merges the closest clusters.

2. Divisive (Top-Down): Starts with all points in a single cluster and recursively splits it into smaller clusters.

Process (Agglomerative Example):

1. Start: Treat each data point as its own cluster.

2. Merge: Calculate & : distance (similarity) between all pairs of clusters, then merge

3. repeat: the steps are repeated

In short, 1. Start: Treat each data point as its own cluster.

2. Merge: Calculate the distance (similarity) between all pairs of clusters, then merge the two closest clusters.

3. Repeat: Continue merging the closest clusters until all points are in a single cluster or a stopping criterion is met.

Example:

Consider a dataset of 5 points: A, B, C, D, E with their respective distances.

1. Step 1: Start with individual clusters: {A}, {B}, {C), {D}, {E}.

2. Step 2: Merge the closest pair, say {A} and {B}, forming {A, B}.

3. Step 3: Now, calculate the distances between {A, Bj, {C {D}, {E}, and merge the closest pair again.

4: Repeat until one cluster remains.

A dendrogram (tree-like diagram) is typically used to visualize the hierarchical

structure of the clusters.

Q6)c. Micro-Average Precision and Recall, Micro-Average F-score.

Ans: Micro-Average Precision, Micro-Average Recall, and Micro-Average F-score are metrics used for multi-class classification

problems, particularly when the classes are imbalanced.

Micro-Average Precision and Recall:

* Micro-Average Precision: Calculated by considering the total true positives, false positives, and false negatives across all classes, rather than calculating precision for each class individually.

* Micro-Average Recall: Similar to precision, it aggregates true positives and false negatives across all classes.

Micro-Average F-score:

* Micro-Average F-score: The harmonic mean of micro-average precision and recall, offering a single score that balances both metrics.

* Formula: Micro-F1 = 2 x Micro-Precision × Micro-Recall Micro-Precision + Micro-Recall

Summary:

* Micro-Average metrics aggregate across all classes, focusing on global performance rather than individual class performance.

* Micro-Average F-score combines precision and recall into a single metric, useful when there are imbalanced class

distributions.

$$\text{Precision}_{\text{Class N}} = \frac{\text{True Positives}_{\text{Class N}}}{\text{True Positives}_{\text{Class N}} + \text{False Positives}_{\text{Class N}}}$$

Source: Author

$$\text{Recall}_{\text{Class N}} = \frac{\text{True Positives}_{\text{Class N}}}{\text{True Positives}_{\text{Class N}} + \text{False Negatives}_{\text{Class N}}}$$

Q7)a. Explain building blocks of RBF networks.

Ans: Recurrent Neural Networks (RNNs) are a type of neural network designed for sequential data. Unlike traditional feedforward networks, RNNs have connections that loop back on themselves, allowing them to maintain a "memory" of previous inputs. This makes them particularly useful for tasks like time series prediction, natural language processing, and speech recognition.

Key Features:

* Memory: RNNs can use information from previous time steps to influence the current prediction.

* Shared Weights: The same weights are applied across different time steps, making RNNs efficient for handling sequences of varying lengths.

Example: Predicting the Next Word in a Sentence

Suppose you're training an RNN to predict the next word in a sentence.

1. Input Sequence: "I love programming"

2. The RNN takes the first word "I" as input, processes it, and updates its internal state.

3. It then takes "love" as input, updates the state again, and so on.

4. At each step, the RNN uses its memory (the internal state) from previous words

to help predict the next word (e.g.,

"programming").

Advantages:

Good for sequential or time-dependent data (e.g., text, speech, financial data).

* Can handle input sequences of varying lengths.

Disadvantages:

* Vanishing Gradient Problem: RNNs can struggle to capture long-term dependencies in data.

* Training Complexity: RNNs can be difficult to train, especially on long sequences.

Q7)b. what are different activation functions in NN?

Ans: Here are some commonly used activation functions in neural networks:

1. Sigmoid:
   Formula:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Outputs values between O and 1, often used in binary classification.

2. Tanh (Hyperbolic Tangent) : outputs values between
   -1 and 1 is a scaled version of sigmoid function.

- Formula: $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$

* 3. ReLU (Rectified Linear Unit):

* Outputs the input it positive, otherwise zero. It's the most widely

used activation function.

* Formula: ReLU(x) = max(0, x)

4. Leaky ReLU:

* A variant of ReLU that allows a small, non-zero gradient for negative inputs.

* Formula: Leaky ReLU(x) = max(ax, x), where a is a small constant.

5. Softmax:

* Used for multi-class classification, it converts logits into probability distributions over multiple classes.

6. ELU (Exponential Linear Unit):

* Similar to ReLU but allows negative outputs, which helps with training stability.

- Formula: $\mathrm{ELU}(x) = x$ if $x > 0$;
  $\alpha(e^x - 1)$ if $x \leq 0$

7. Swish:

* A newer activation function that performs better than ReLU in some cases.

*
- Formula: $\mathrm{Swish}(x) = x \cdot \sigma(x)$,

where o (x) is the sigmoid function.

Q7)c. What is multilayer perceptron? Describe with diagram.

Ans: A Multilayer Perceptron (MLP) is a type of feedforward neural network consisting of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer. MLPs are widely used for classification and regression tasks.

Structure of MLP:

1. Input Layer: The first layer that receives the input features.

2. Hidden Layers: One or more layers where computations are performed, and non-linear activation functions (e.g.,

ReLU, Sigmoid) é v ipplied.

2 output layer: The final layer that produces the output, typically using an activation function like Softmax (for classification) or linear (for regression).

How it works:

Data is passed from the input layer to the first hidden layer where it is processed.

Finally, the output layer produces the final prediction.

Advantages:

* Can model complex relationships through non-linear transformations.

* Suitable for supervised learning tasks.

Disadvantages:

* Requires large amounts of data and computational resources, especially for deep networks.