M.TECH II-I SEM END EXAM (Regular) January-2026

SCHEME OF EVALUATION - MACHINE LEARNING

PART-A

SL NO	THEORY	MARKS	TOTAL
a.	What are the advantages and disadvantages of using a small K value in K-nearest neighbors?	1	1
b.	What is the concept of ranking in machine learning?	1	1
c.	List out the applications of matrix completion	1	1
d.	What are mixture models in unsupervised learning? Discuss.	1	1
e.	What is an F1-Score? How to use it?	1	1
f.	Define overfitting and underfitting?	1	1
g.	What is feature representation learning?	1	1
h.	Provide an example where sparse models can be applied.	1	1
i.	What are the challenges in handling IoT Data?	1	1
j.	What made scikit learn very popular in the machine learning community?	1	1

PART-B

SLNO	THEORY	MARKS	TOTAL
2.	What is the Naive-Bayes algorithm, and how does it work in classification tasks? what assumptions does it make about the data?	10	10
3.	Explain the basic principles of support vector machines and how they work in binary classification.	10	10
4.	Describe the K-means clustering algorithm and its steps. How does the choice of the initial centroids impact the convergence of the K-means algorithm?	10	10
5.	How does PCA help in reducing the dimensionality of a dataset while retaining information?	10	10
6.	Describe the concept of cross-validation. provide examples of situations where K-fold cross-validation would be preferable over a simple train-test split	10	10
7.	Compare and constrast Random forest with bagging and boosting in terms of their underlying machanisms.	10	10
8.	Explain the concept of sparse modeling and its significance in feature selection.	10	10
9.	a) Explain the key characteristics of time series data	5	10
	b) What are the challenges associated with modeling time series data compared to static data?	5	
10.	Identify and analyze two challenges specific to implementing classification methods for IoT applications.	10	10
11.	Choose one recent trent in machine learning and discuss its applications in a specific industry or domain	10	10

PART-A

1. a) What are the Advantages and disadvantages of using a Small K in KNN ?

Advantages : Captures local patterns, low bias, flexible decision boundary.

Disadvantages : Sensitive to noise and outliers, high variance, may overfit.

b) What is the concept of ranking in machine learning ?

Ranking is the task of ordering items based on relevance or preference, often used in search engines, recommendation systems, and information retrieval.

c) List out the Applications of matrix completion ?

* Recommendation Systems (eg., Netflix, Amazon)
* Image inpainting
* collaborative filtering
* Sensor data recovery

d) What are the Mixture models in unsupervised learning ? Discuss.

Mixture models assume data is generated from multiple underlying probability distributions. Each data point belongs to a latent cluster with a certain probability (e.g., Gaussian Mixture Models).

e) What is an F1-score ? How to use it.

F1-score is the harmonic mean of precision and Recall.

$F1 = 2PR/P+R$

used when class imbalance exists and both precision and recall are important.

f.) Define overfitting and underfitting?

overfitting: Model learns noise, performs well on training but poorly on test data.

underfitting: Model is too simple, performs poorly on both training and test data.

g.) What is feature representation learning?

It is the process of automatically learning useful features from raw data instead of manual feature engineering (e.g., embeddings, deep learning).

h.) provide an Example where sparse models can be applied?

* Text classification using bag-of-words.
* LASSO regression
* Recommendation Systems with sparse user-item matrices.

i) What are the challenges in handling IOT data?

* High volume and velocity
* Noisy and incomplete data
* Real-time processing requirements
* Security and privacy issues.

j) What made Scikit-learn very popular in the machine learning community?

* Simple and consistent
* Rich collection of ML algorithms
* Excellent documentation and community support
* Easy integration with Numpy and pandas

PART - B

2. What is the Naive Bayes Algorithm and how does it work in classification tasks? What assumptions does it make about the data?

Introduction:

Naive Bayes is a probabilistic supervised learning algorithm based on Bayes' Theorem. It is widely used for classification tasks, especially in text classification, Spam detection, and sentiment analysis.

Bayes' Theorem:

$$P(C|x) = P(x|C)\,P(C) / P(x)$$

Where:

$P(C|x)$: posterior probability of class given features

$P(C)$: prior probability of class

$P(x|C)$: Likelihood

$P(x)$: Evidence

Working of Naive Bayes

1. Calculate prior probabilities for each class.
2. compute likelihood of features given the class
3. Apply the naive independence assumption.
4. compute posterior probabilities for all classes.
5. Assign the class with maximum posterior probability.

Types of Naive Bayes classifiers:

→ Gaussian Naive Baye - continuous features

→ Multinomial Naive Bayes - text and word counts

→ Bernoulli naive Bayes - binary features.

Advantages:

→ Simple and easy to implement

→ works well with high-dimensional data

→ Fast training and prediction

→ Effective for text classification

Limitations:

→ Independence assumption is often unrealistic

→ Zero-frequency problem

→ poor performance when features are highly correlated

Assumptions of naive Bayes:

→ features are conditionally independent given the class

→ All features contribute equally and independently

→ Data follows an assumed distribution.

Applications:

→ Spam filtering

→ Sentiment analysis

→ Document classification

→ Medical diagnosis

3. Explain the Basic principals of support vector Machines (SVM) and how they work in Binary classification?

Introduction:

Support vector Machine (SVM) is a Supervised learning algorithm used for classification and regression. It aims to find an optimal separating hyperplane between classes.

Basic Concept:

SVM finds the hyperplane that maximizes the margine between two classes.

Margin is the distance between the hyperplane and the nearest data points (sv).

Support vectors:

- Data points closest to the hyperplane
- They determine the position and orientation of the decision boundary.

Linear SVM:

for linearly separable data:

$w \cdot x + b = 0$

Where:

w = weight vector

b = bias.

Objective:

$\min 1/2 \|w\|^2$

Non-linear SVM and kernel trick:

for non-linearly separable data, SVM uses kernal functions to map data into higher dimensions.

Common kernals:

linear

polynomial

Radial Basis function (RBF)

Sigmoid.

Soft Margin and Regularization:

* Introduces slack variables to allow misclassification

* Regularization parameter C controls trade-off between margin and classification

Advantages:

* Effective in high-dimensional spaces
* Robus to overfitting
* works well with small datasets.

Disadvantages:

* computationally expensive for large datasets
* choice of kernel and parameters is critical.
* not very interpretable.

Applications:

→ Text classifcation
→ Image recognition
→ Bioinformatics
→ face detection.

---

4. Describe K-Means clustering Algorithm and its steps. How does the choice of Initial centroids impact the convergence of the k-Means algorithm?

Introduction:

K-Means is an unsupervised clustering algorithm that partitions a dataset into k distanct clusters by minimizing intra-cluster variance.

Objective function:

The goal of k-Means is to minimize:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Where :

C_i : i-th cluster

μ_i : centroid of cluster

Steps of K-Means Algorithm:

* choose the number of clusters k
* Initialize k centroids randomly.
* Assign each data point to the nearest centroid
* Recalculate centroids as the mean of assigned points.
* Repeat steps 3 and 4 until centroids converge or assignments do not change.

Convergence of K-Means:

* The algorithm converges when centroids stop changing.
* It always converges to a local minimum, not necessarily a global minimum.

Impact of Initial centroids:

* poor initialization can lead to :
  - → slow convergence
  - → suboptimal clustering
  - → Empty clusters
* Different initial centroids may produce different final clusters.

Improved Initialization Techniques:

* K-Means ++ : chooses initial centroids that are far apart.
* Multiple random restarts to select best clustering.

Advantages:

* simple and computationally efficient.
* scales well for large datasets.

Limitations:

* sensitive to initialization
* Requires pre-specifying k
* performs poorly on non-spherical clusters.

5. How does PCA help in reducing the Dimensionality of a dataset while Retaining Information?

Introduction:

Principal Component Analysis (PCA) is a linear dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving maximum variance.

Objective of PCA:

* Reduce dimensionality
* Remove redundancy
* Improve computational efficiency
* preserve important information.

Steps Involved in PCA:

* Standardize the dataset
* Compute Covariance matrix
* calculate eigenvalues and eigenvectors
* select top k-eigenvectors

* project data onto the new feature space

Principal components:

* orthogonal directions capturing maximum variance.
* first principal component captures highest variance, second captures next highest, and so on.

Variance Retention:

* Eigenvalues represent the amount of variance captured.
* Choose components that retain a desired percentage (eg., 95%) of total variance.

How PCA Retains Information:

* By preserving directions with maximum variance.
* Eliminates noise and correlated features.
* Minimizes reconstruction error.

Advantages:

* Reduces overfitting.
* Improves visualization.
* speeds up training of ML models.

Limitations:

* loss of interpretability
* works only on linear relationships
* sensitive to feature scaling.

Applications:

* Image compression.
* Face recognition.

* Noise reduction
* feature extraction.

6. Describe the concept of cross-validation. provide examples of Situations where k-fold cross-validation would be preferable over a simple train-test split

Cross-validation:

Cross-validation is a model evaluation technique used to assess how well a machine learning model generalizes to unseen data. Instead of using train-test split, the dataset is divided into multiple subsets, and the model is trained and tested multiple times.

The main goal of cross-validation is to reduce overfitting, use data efficiently, and obtain a reliable estimate of model performance.

K-Fold cross-validation:

In K-fold cross-validation, the dataset is divided into k-equal-sized folds:

1. The model is trained on k-1 folds
2. Tested on the remaining 1 fold
3. This process is repeated k times, each fold acting as the test set once
4. The final performance is the average of all k results

Common values of K are 5 or 10.

Steps of K-fold cross-validation:

1. Shuffle the dataset randomly.
2. split the dataset into k folds
3. Train the model on k-1 folds

* Final output is obtained by majority voting or averaging.

Key features:

* Reduces variance
* works well with high-variance models
* Example : Bagged Decision Trees.

Boosting:

* Models are trained sequentially
* Each new model focuses more on previously misclassified samples
* Samples are weighted differently.

key features:

* Reduces bias and variance
* Sensitive to noise and outliers
* Examples: AdaBoost, Gradient Boosting, XGBoost

Random forest:

Random forest is an extension of baggin with additional randomness:

* Uses bootstrap Sampling
* At each split, selects a random subset of features
* Builds multiple decision trees independently.

Key features:

* Reduces correlation among trees
* Improves accuracy and robustness
* Handles high-dimensional data well.

4. validate it on the remaining fold
5. Repeat for all folds
6. Compute the average accuracy/error).

Why k-Fold cross-validation is Better than Train-Test split

Train-Test Split	k-Fold Cross-Validation
* uses data only once	* uses data efficiently
* Results depend on split	* More stable results
* High variance	* Lower variance
* Less reliable	* More reliable.

Situations where k-Fold Cross-Validation is preferable

1. Small datasets - Every data point is important
2. Model comparison - fair evaluation of multiple models
3. Hyperparameter tuning - used with GridSearch or RandomSearch
4. Imbalanced datasets - Reduces based evaluation
5. High-variance models (eg., Decision Trees)
6. Research and academic experiments.

7. Compare and contrast Random forest with bagging and boosting in terms of their underlying mechanisms.

Ensemble learning: Ensemble methods combine multiple models to improve accuracy, stability, and generalization.

Bagging (Bootstrap Aggregating)

* Multiple models are trained on random boostrap samples
* Models are trained independently

Mathematical Intuition:

Sparsity is encouraged using regularization techniques, especially:

* L1 regularization (Lasso)

Objective function (example): $\min(\text{loss} + \lambda \Sigma |w_i|)$

Techniques used in Sparse Modeling:

1. Lasso Regression (L1 regularization)
2. Elastic Net (L1+L2)
3. Sparse PCA
4. Compressed sensing
5. Sparse coding.

Significance in feature selection:

1. Automatic feature selection:- irrelvent features get zero weight
2. Reduced dimensionality - fewer features used
3. Improved generalization - less overfitting
4. Model interpetability - eaisey to understand
5. Computational efficiency - faster training and prediction
6. Handles high-dimensional data (e.g., text, gene data)

Applications:

* Text classification (bag-of-words)
* Image processing
* Bioinformatics
* Recommendation Systems.

Aspect	Bagging	Boosting	Random Forest
* Training	parallel	sequential	parallel
* Data Sampling	Bootstrap	weighted	Bootstrap
* Focus on Errors	NO	Yes	NO
* Feature Randomness	NO	NO	yes
* overfitting control	Reduces variance	Reduces bias	Reduces variance
* Noise Sensitivity	Low	High	low

8. Explain the concept of Sparse modeling and its significance in feature selection.

Sparse Modeling:

Sparse modeling is a machine learning approach in which the model is trained using only a small subset of relvant features, while the remaining features are assigned zero or near-zero weights. The main idea is that not all features contribute equally to prediction, and many can be safely ignored without losing important information.

Key Idea:

Given a large number of features, Sparse models:

* select only important features
* Improve interpretability
* Reduce overfitting.

9. a) Explain the key characteristics of time series data. 8

Time series data consists of observations recorded over time at regular or irregular intervals. The order of data points is crucial.

Key characteristics :

1. Trend

   long-term increase or decrease in data

   Example : Rising temperature over years.

2. Seasonality

   Repeating patterns at fixed intervals.

   Example : Sales increasing during festivals.

3. Cyclic patterns

   Fluctuations without fixed period

   Example : Economic cycles

4. Stationarity :

   Statistical properties (mean, variance) remain constant over time

5. Autocorrelation :

   Dependency between current and past values.

Examples :

* Stock prices
* weather data
* Sensor readings.

9. b) What are the challenges associated with modeling time series data compared to static data?

Modeling time series data is more challenging than static data because observations are time-dependent and their order cannot be ignored. The main challenges are explained below:

1. Temporal Dependancy:

In time series data, current observations depends on past values. Traditio-nal machine learning models assume independent samples, which is not valid for time series.

2. Non-Stationarity:

Time series data often exhibits changing mean, variance, or seasonality over time. Many models require Stationarity, so preprocessing techniques like differencing or detrending are needed.

3. Seasonality and Trend Components:

The presence of seasonal patterns and trends complicates modeling, as these patterns must be identified and modeled separately for accurate forecasting.

4. Missing and Irregular Data:

Time series datasets may have missing values, irregular time intervals, or sensor failures, which affect model reliability and require special handling.

5. Noise and outliers:

Time series data is often noisy due to measurement errors or external disturbances. outliers can severely affect forecasting accurancy.

6. Long-Term Dependencies:

Capturing dependencies over long time horizons is difficult, especially for traditional models. Advanced models like LSTM and GRU are often required.

7. Limited Data Availability:

Unlike static datasets, time series data cannot be easily augmented, and historical data may be limited, affecting model performance.

8. Concept Drift:

The underlying data generation process may change over time, causing previously learned patterns to become invalid.

9. Evaluation complexity:

Standard random train-test splits cannot be used. Time-aware validation techniques are required to avoid data leakage.

10. computational complexity:

Time series models often require extensive preprocessing and parameter tuning, increasing computational cost.

---

10. Identify and analyze two challenges specific to implementing classification methods for IoT applications.

Introduction:

The Internet of Things (IoT) consists of a large number of interconnected devices such as sensors, actuators, and smart objects that continuously generate data. Classification method play a crucial role in IoT applications like Smart healthcare, Smart cities, industrial monitoring, and intrusion detection. However, implementing classification algorithms in IoT environments poses several challenges due to

System constraints and data characteristics

This answer discusses two major challenges:

1. Resource constraints of IoT devices
2. Data quality, heterogeneity, and scalability.

Challenge 1: Resource Constraints of IoT Devices

Description:

Most IoT devices operate with limited computational power, memory, storage, and battery life. Advanced classification algorithms such as deep neural networks, ensemble models, and kernel-based methods are computationally intensive and require significant memory

Impact on classification Methods:

* complex models cannot be deployed directly on edge devices.
* High latency occurs if data is transmitted to cloud servers.
* Increased energy consumption reduces device lifetime
* Real-time classification becomes difficult.

Analysis:

* Lightweight classification such as Naive Bayes, Decision Trees, and K-NN (with small K) are preferred.
* Edge computing is used to process data closer to devices
* Model compression techniques such as pruning, quantization, and knowledge distillation help reduce model size
* A trade-off must be made between accuracy, latency, and energy efficiency.

Example:

In wearable health monitoring systems, continuous ECG & heart-rate classification must be performed using energy-efficient models to avoid frequent battery replacement.

Challenge 2: Data quality, Heterogeneity, and Scalability:

Description:

IoT environments generate massive volumes of data from heterogeneous sources, such as temperature sensors, cameras, motion detectors, and Gps modules. This data is often:

- Noisy
- Incomplete
- unlabeled or weakly labeled
- Generated at different sampling rates

Impact on classification methods:

* Noisy and missing data leads to incorrect predictions
* Feature extraction becomes complex due to diverse data fromats.
* Class imbalance is common
* Scalability issues arise with increasing number of devices

Analysis:

* Extensive data preprocessing is required.
* Robust classifiers and ensemble methods help handle noise
* semi-supervised and online learning techniques are used due to lack of labeled data.
* Distributed and cloud-based classification frameworks improve scalability.

Example:

In Smart city traffic monitoring, data from cameras, sensors, and Gps systems must be integrated and classified despite missing values and varying data formats.

Additional considerations:

* privacy and security issues: Sensitive IoT data must be protected during classification.
* concept drift: Data distribution changes over time, requiring adaptive models
* Real-time constraints: Some applications requires instant classification results Top of form

11. choose one recent trend in machine learning and discuss its applications in a specific industry or domain.

Introduction:

Machine learning has evolved rapidly in recent years, leading to the emergence of several new trends that enhance model performance, scalability, and real-world application. One of the most significant recent trends is federated learning (FL), which enables collaborative model training across distributed devices while preserving data privacy.

This answer discusses federated learning and its applications in the Healthcare industry.

Recent Trend: federated learning

Definition:

Federated learning is a decentralized machine learning approach in which multiple devices or organizations collaboratively train a shared global model without exchanging raw data. Instead, each participant trains the model locally and shares only model updates with a central server.

working principle:

1. A global model is initialized at a central server
2. The model is sent to participating devices or institutions
3. Each participant trains the model on local data
4. only model updates are sent back
5. The server aggregates updates to improve the global model.
6. Steps repeat until convergence.

Applications in the Healthcare Industry:

1. Medical Image Analysis:

Hospitals can collaboratively train models for:

* Cancer detection
* X-ray and MRI classification
* Tumor segmentation

without sharing patient data, thus ensuring data privacy.

2. Disease prediction and Diagnosis:

Federated learning enables:

* Training predictive models on patient health records
* Early detection of diseases such as diabetes and heart disorders
* Improved diagnostic accuracy across hospitals.

3. Wearable Health Monitoring:

Data from smartwatches and medical sensors is used to:

* Classify abnormal heart rhythms
* Detect sleep disorders.
* Monitor chronic conditions

All while keeping user data on personal devices.

4. Drug Discovery and Clinical Research:

pharmaceutical companies can:

* collaborate on model training
* Analyze distributed clinical trial data
* Reduce data-sharing risks.

Advantages of federated learning

1. Data privacy and Security
2. Compliance with healthcar regulations (HIPAA, GDPR)
3. Reduced data transfer costs
4. Enables collaboration across institutions
5. Scales well with distributed data.

challenges:

* communication overhead
* Data heterogeneity across participants
* Model aggregation complexity
* Security risks such as model poisoning.

Future Scope:

* Integration with edge computing
* secure aggregation techniques
* personalized federated models
* wider adoption in sensitive domains.