DEPARTMENT OF INFORMATION TECHNOLOGY

UNIT TEST – I SOLUTION

Class: BEIT **Semester: VII**

Date: 29/08/2024 **Time: 2:00-3:30 PM** **Max marks: 40**

Subject: HAIMLC701- AI&ML in Healthcare

Note the following instructions

1. **Attempt all questions.**
2. **Draw neat diagrams wherever necessary.**
3. **Write everything in ink (no pencil) only.**
4. **Assume data, if missing, with justification.**

Q1	Attempt any two	Marks	CO
A	Explain the four distinct categories of AI with an example for each. **Four distinctive categories of AI:** 1. Reactive Machines • most basic AI • respond in a current scenario, relying on taught or recalled data to make decisions in their current state • perform the tasks well for what they are designed, but not others. • Do not store memories or past experiences for future action • Example-Deep Blue, the chess-playing IBM supercomputer, was a reactive machine, able to make predictions based on the chessboard at that point in time. • Deep Blue beat world champion chess player Garry Kasparov in 1996. • Google's AlphaGo 2. Limited Memory: Systems That Think and Act Rationally • works on the principle of limited memory • uses both preprogrammed knowledge and subsequent observations carried out over time • During observations, the system looks at items within its environment and detects how they change and then makes necessary adjustments • E.g. Self-driving cars are one of the best examples of Limited Memory systems. These cars can store recent speed of nearby cars, the distance of other cars, speed limit, and other information to navigate the road 3. Theory of Mind: Systems That Think Like Humans	[5]	CO1

	• This kind of AI requires an understanding that the people and things within an environment can also alter their feelings and behaviors • Theory of Mind AI should understand the human emotions, people, beliefs, and be able to interact socially like humans. • This type of AI machines are still not developed, but researchers are making lots of efforts and improvement for developing such AI machines • Although such AI is presently limited, it could be used in caregiving roles such as assisting elderly or disabled people with everyday tasks • Theory of mind AI can attempt to understand people's intentions and predict how they may behave. 4. Self-Aware AI: Systems That Are Humans • The future of Artificial Intelligence • These machines will be super intelligent, and will have their own consciousness, sentiments, and self-awareness • These machines will be smarter than human mind • Self-Awareness AI does not exist in reality still and it is a hypothetical concept		
B	Explain in brief the following applications of AI in healthcare. (i) Follow up care (ii) Prediction **Follow-Up Care** Hospital readmittance is a huge concern in healthcare. Doctors, as well as governments, are struggling to keep patients healthy, particularly when returning home following hospital treatment. Organizations such as NextIT have developed digital health coaches, similar to a virtual customer service representative on an e-commerce site. The assistant prompts questions about the patient's medications and reminds them to take medicine, queries them about their condition symptoms, and conveys relevant information to the doctor. **Prediction** Technologies already exist that monitor data to predict disease outbreaks. This is often done using real-time data sources such as social media as well as historical information from the Web and other sources. Malaria outbreaks have been predicted with artificial neural networks, analyzing data including rainfall, temperature, number of cases, and various other data points.	[5]	CO1

*************** All The Best ******************

C	Elaborate on the following challenges of AI & ML in the Healthcare domain. (i) Bias (ii) Data Governance **Bias** A significant problem with learning is bias. As AI becomes increasingly interwoven into our daily lives—integrated with our experiences at home, work, and on the road—it is imperative that we question how and why machines do what they do. Within machine learning, learning to learn creates its own inductive bias based on previous experience. Essentially, systems can become biased based on the data environments they are exposed to. It's not until algorithms are used in live environments that people discover built-in biases, which are often amplified through real-world interactions. This has expedited the growing need for more transparent algorithms to meet the stringent regulations on drug development and expectation. Transparency is not the only criteria; it is imperative to ensure decision- making is unbiased to fully trust its abilities. People are given confidence through the ability to see through the black box and understand the causal reasoning behind machine conclusions. **Data Governance** With this comes the topic of data governance. Medical data is personal and not easy to access. It is widely assumed that the general public would be reluctant to share their data because of privacy concerns. However, a Wellcome Foundation survey conducted in 2016 on the British public's attitude to commercial access to health data found that 17% of people would never consent to their anonymized data being shared with third parties. Adhering to multiple sets of regulation means disaster recovery and security is key, and network infrastructure plays a critical role in ensuring these requirements can be met. Healthcare organizations require modernization of network infrastructure to ensure they are appropriately prepared to provide the best patient care possible. The case for this is highlighted by the fact that the bulk of NHS computers in 2018 use Internet Explorer 8 as their default Internet browser - which was first launched almost a decade ago.	[5]	CO1
D	Explain the use case "Electronic Health Records". **Electronic Health Records** EHRs haven't quite come to fruition as yet. The idea is theoretically simple: that every patient has a digital health record consisting of their details, demographics, medical history, allergies, clinical results, and so forth. Records can be shared, with patient consent, via secure computer systems and are available for healthcare providers from both	[5]	CO1

	public and private sectors. Each record comprises one modifiable file, which mean that doctors can implement changes over time with no danger of data replication or inconsistencies. EHRs make perfect sense; however, complete implementation across a nation is proving a task. In the United States, up to 94% of hospitals use EHRs according to HITECH research. Europe is further behind. A European Commission directive has set the task of creating a centralized European health record system by 2020. In the United States, Kaiser Permanente has implemented a system that shares data across all their facilities and made it easier to use EHRs. A McKinsey report highlighted how the data sharing system achieved an estimated $1 billion in savings as the result of reduced office visits and lab tests. The data sharing system improved outcomes in cardiovascular disease. The EHR is evolving into the blockchain, which seeks to decentralize and distribute access to data.		
Q2	**Attempt any two**		
A	Compare and Contrast the Ensemble Techniques: Bagging, Boosting and Stacking. Bagging, Boosting, and Stacking are ensemble learning techniques used to improve the performance and robustness of machine learning models. Here's a comparison of these methods: **1. Bagging (Bootstrap Aggregating)** **Key Idea**: • Bagging aims to reduce variance by training multiple models independently on different subsets of the data and then aggregating their predictions. **Process**: • Randomly sample subsets of the training data (with replacement). • Train a base model (like Decision Trees) on each subset. • Combine the predictions by averaging (for regression) or voting (for classification). **Advantages**: • Reduces variance and helps prevent overfitting. • Works well with high-variance models like Decision Trees. **Disadvantages**: • May not significantly improve performance if the base model has low variance or high bias. **Popular Algorithms**: • Random Forest (a type of Bagging using Decision Trees). **2. Boosting** **Key Idea**:	[10]	CO2

- Boosting reduces bias by combining multiple weak learners to create a strong learner. Each model tries to correct the errors of its predecessor.

Process:
- Train a model on the entire dataset.
- Assign higher weights to misclassified examples.
- Train the next model focusing more on the difficult cases.
- Combine models typically through weighted voting or averaging.

Advantages:
- Focuses on difficult examples, leading to better overall performance.
- Reduces bias and can achieve high accuracy.

Disadvantages:
- Can be prone to overfitting if not carefully regularized.
- Training is sequential, making it slower than Bagging.

Popular Algorithms:
- AdaBoost, Gradient Boosting Machines (GBM), XGBoost, LightGBM, CatBoost.

3. Stacking (Stacked Generalization)

Key Idea:
- Stacking combines multiple models by training a "meta-model" to aggregate the predictions of the base models.

Process:
- Train different base models on the same dataset.
- Use the predictions of these models as inputs for a meta-model.
- Train the meta-model to make the final predictions.

Advantages:
- Can capture complex patterns by leveraging the strengths of different models.
- More flexible, as base models can be of different types.

Disadvantages:
- Computationally expensive.
- Requires careful design to avoid overfitting, especially when the base models are highly correlated.

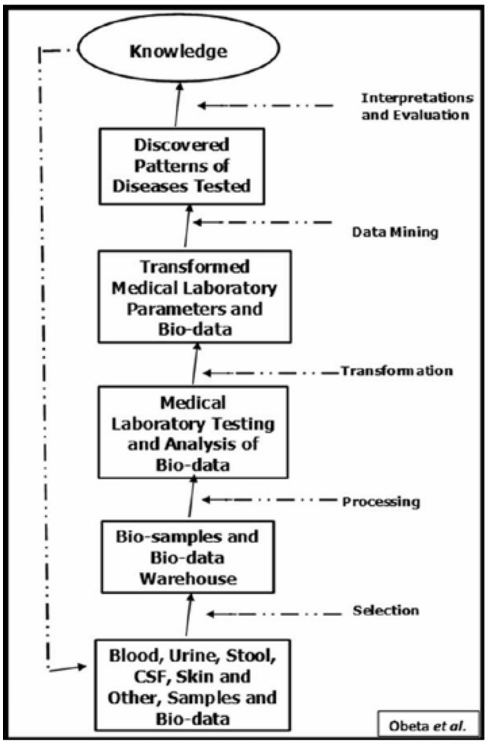
Feature	Bagging	Boosting	Stacking
Goal	Reduce variance	Reduce bias	Combine diverse models
Model Training	Independent models	Sequential models	Layered model approach
Performance	Improves stability	Improves accuracy	Combines strengths
Risk	Lower risk of overfitting	Higher risk of overfitting	Risk of overfitting if not careful
Speed	Faster (parallel training)	Slower (sequential training)	Slower (complex training)
Common Algorithms	Random Forest	AdaBoost, XGBoost	Custom implementations

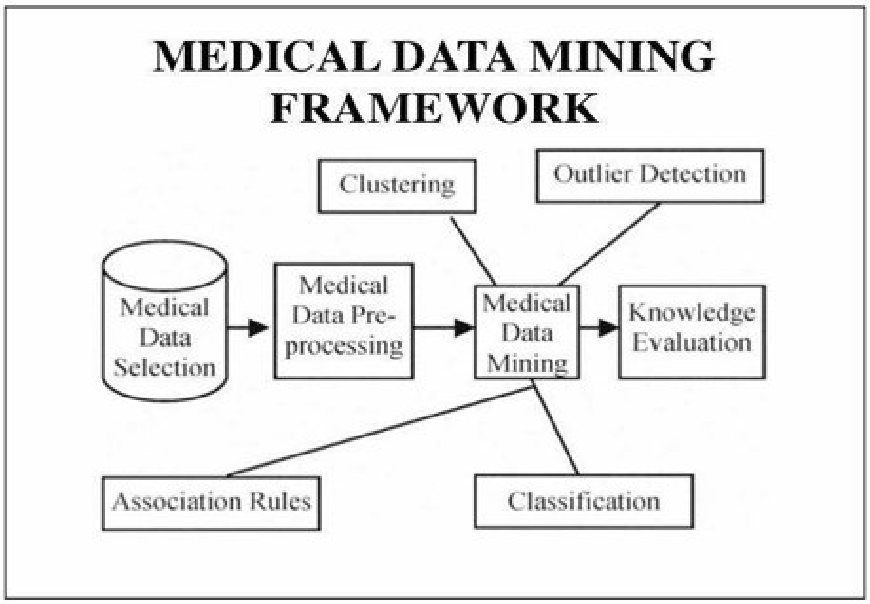
B	Demonstrate the steps involved in medical data mining and explain any two techniques with appropriate example.	[10]	CO2

Knowledge Discovery and Data Mining in Healthcare

Data mining (knowledge discovery from data)

• Extraction of interesting (non-trivial, implicit previously unknown, and potentially useful) patterns or knowledge from huge amount of data





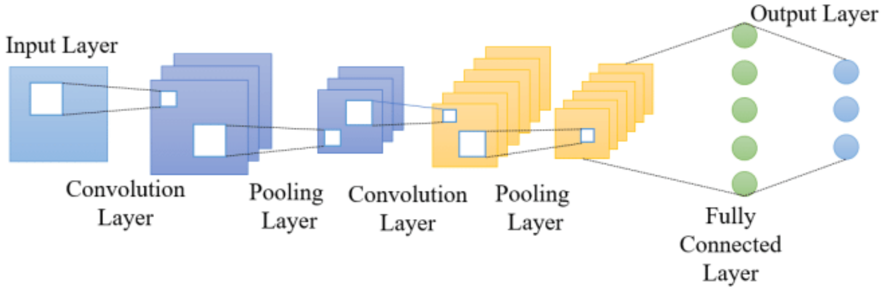
Data Mining Techniques in Healthcare

	• Medical data has a lot of information buried within it that will reveal patterns relating to successes and failures in clinical operations **Association or relationship analysis** • If a group of patients with specific symptoms is steadily associated with certain prescribed medications they acquire in pharmacies during a preset season, pharmacists can use this information to manage their stock **Sequence analysis** • Consequential flows of facts or eventsall patients of the group also share similar lifestyles, chronic diseases, or other health features. With this knowledge, physicians can offer preventive care **Classification** • Cases can be compared with each other to be verified as falling within a certain class, to identify differences and apply necessary algorithms and protocols, or to screen out and readdress unmatching data		
C	Make use of the architecture of Convolutional Neural Network to illustrate its usage in disease diagnosis with a neat diagram. Convolutional Neural Networks (CNNs) are widely used in disease diagnosis, particularly in medical imaging, due to their ability to detect complex patterns in images. CNNs can analyze medical images like X-rays, MRIs, CT scans, and histopathological slides to help identify signs of disease, including cancers, lung diseases, brain anomalies, and other conditions. **Architecture and Usage of CNN in Disease Diagnosis** A typical CNN used for disease diagnosis in medical imaging follows this structure: 1. **Input Layer**: o The network receives an input image (e.g., an X-ray or MRI scan) that is typically preprocessed to standard dimensions. 2. **Convolutional Layers**: o The CNN applies multiple convolutional filters to scan over regions of the image, extracting important features like edges, textures, and patterns that are useful for identifying abnormalities. o Each convolutional layer enhances different features that represent structures indicative of a disease. 3. **Pooling Layers**: o Pooling layers reduce the spatial dimensions of the image, keeping only the most important features and	[10]	CO2

reducing computation. Max pooling is commonly used to select the highest feature values within a region.
    - Pooling also helps the model generalize better by making it invariant to slight shifts or rotations in the image.
4. **Additional Convolution and Pooling Layers** (Optional):
    - Deeper CNNs often stack more convolution and pooling layers, enabling the model to capture more complex patterns and hierarchical features.
5. **Flattening Layer**:
    - After feature extraction, the high-dimensional output from the convolutional layers is flattened into a one-dimensional vector to prepare for the fully connected layers.
6. **Fully Connected (Dense) Layers**:
    - The dense layers learn non-linear combinations of the extracted features, enabling more precise classification.
    - These layers play a crucial role in distinguishing between normal and abnormal regions or between different types of diseases.
7. **Output Layer**:
    - The final layer outputs the probability or classification label, identifying the presence or absence of disease, or the specific type of disease if it's a multiclass problem.
    - Common loss functions like cross-entropy are used to measure prediction accuracy.

Example of Disease Diagnosis Using CNN

A CNN might be trained on a dataset of chest X-rays to diagnose pneumonia. During training, the network learns features that distinguish pneumonia-affected lung areas from healthy ones, such as tissue opacity and structure distortions.



Q3	**Attempt any one**		
A	Make use of the given confusion matrix of a classifier which detects TB in patients with two classes: the disease is present (class Y) or absent (class N). Explain all the terms and formula required and calculate the following performance metrics:(i) Accuracy (ii) Precision (iii) Specificity (iv) Recall (v) F-measure	[10]	CO3

	Predicted		
Actual		**Y**	**N**
	Y	910	90
	N	70	300

SET-1

Actual	Predicted Y	Predicted N
Y	910	90
N	70	300

TP - 910 FP - 70

TN - 300 FN - 90

(i) Accuracy $= \frac{TP+TN}{TP+FP+TN+FN} \times 100 = \frac{1210}{1370} \times 100 = 88.3$

(ii) Precision $= \frac{TP}{TP+FP} = \frac{910}{980} = \cancel{92.8}\ 0.928$

(iii) Specificity $= \frac{TN}{TN+FP} = \frac{300}{370} = 0.81$

(iv) Recall $= \frac{TP}{TP+FN} = \frac{910}{1000} = 0.91$

(v) F-measure $= \frac{2TP}{2TP+FP+FN} = \frac{1820}{1820+160} = 0.919$

B	Differentiate Model parameters from Hyperparameters. Demonstrate Grid search and Random search hyper parameter tuning algorithms with an example.	[10]	CO3

Difference between Model Parameters and Hyperparameters

1. Model Parameters:
    - Parameters are values that a machine learning model learns from the training data automatically.
    - They are internal to the model and are essential for making predictions on new data.

- Examples include weights in a neural network, coefficients in linear regression, and decision tree splits.

2. Hyperparameters:
   - Hyperparameters are external configurations set before training the model. They control the model training process and influence the performance and complexity of the model.
   - They are not learned from the data, and their values need to be manually set or optimized.
   - Examples include learning rate, number of epochs, batch size in neural networks, depth of decision trees, and number of neighbours in k-nearest neighbours.

Hyperparameter Tuning Algorithms: Grid Search and Random Search

1. Grid Search
Grid Search is a hyperparameter tuning method where we define a set of possible values for each hyperparameter, and the algorithm evaluates all possible combinations of these values. While it provides the optimal combination within the search space, it can be computationally expensive, especially with a large set of hyperparameters.

2. Random Search
Random Search is a method where random combinations of hyperparameters are selected from a predefined search space. This approach does not evaluate every combination, making it faster and more efficient, especially when the search space is large. It is useful for high-dimensional data and when only a subset of hyperparameter combinations yields good results.

Example: Grid Search vs. Random Search for Hyperparameter Tuning

Suppose we are training a Support Vector Machine (SVM) classifier on a dataset. The SVM model has the following hyperparameters:

- C: Regularization parameter
- Kernel: Kernel type (linear, polynomial, RBF)
- Gamma: Kernel coefficient for 'rbf' and 'poly' kernels

1. Grid Search for SVM Hyperparameter Tuning

With Grid Search, we define a set of values for each hyperparameter and evaluate all possible combinations.

Sample Code:

```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Load the data and split it
data = load_iris()
X_train, X_test, y_train, y_test = train_test_split(data.data,
data.target, test_size=0.2, random_state=42)

# Define the parameter grid
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': [0.01, 0.1, 1]
}

# Initialize Grid Search with cross-validation
grid_search = GridSearchCV(SVC(), param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Best parameters from Grid Search
print("Best parameters from Grid Search:",
grid_search.best_params_)
```

Explanation: Grid Search will try all $3 \times 2 \times 3 = 18$ possible combinations of C, kernel, and gamma, and select the best based on the cross-validation results.

2. Random Search for SVM Hyperparameter Tuning

With Random Search, we specify a range of values for each hyperparameter and let the algorithm sample a random combination of these values.

Sample Code:

```
from sklearn.model_selection import RandomizedSearchCV
import scipy.stats as stats
```

	# Define the parameter distribution
	param_dist = {
	'C': stats.uniform(0.1, 10),
	'kernel': ['linear', 'rbf'],
	'gamma': stats.uniform(0.01, 1)
	}
	
	# Initialize Random Search with cross-validation
	random_search = RandomizedSearchCV(SVC(), param_distributions=param_dist, n_iter=10, cv=5, random_state=42)
	random_search.fit(X_train, y_train)
	
	# Best parameters from Random Search
	print("Best parameters from Random Search:", random_search.best_params_)
	
	Explanation: Here, RandomizedSearchCV randomly selects 10 combinations from the parameter distribution rather than evaluating every possible combination. This is often much faster and can still yield an effective model configuration.
	
	In Summary, **Grid Search** evaluates all possible combinations within the defined grid and is best suited for smaller, discrete hyperparameter spaces. **Random Search** samples random combinations from the search space and is ideal for larger or continuous hyperparameter spaces where an exhaustive search is impractical.		