SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANSWER KEY SUBMISSION

Date of Exam & Session	**31/03/2023 FN**	**Category of Exam**	**CLA2**
Course Name	**Applied Machine Learning**	**Course Code**	**18CSE481T**
Name of the Faculty submitting	**Dr. M. Mahasree**	**Date of submission of Answer Key**	**12/04/2023**
Department to which the faculty belongs to	**CSE**	**Total Marks**	**50**

PART A(10x1= 10)
ANSWER ALL THE QUESTIONS

Q.No.	MCQ Questions	Marks	CO	BL	PI
1.	The signal that is used in speech recognition is known as? **(A). Acoustic signal** (B). Electric signal (C). Electromagnetic signal (D). Radar	1	2	1	1.5.1
2.	Select the dominant modality for communication between humans? (A). Hear **(B). Speech** (C). Smell (D). None of these	1	2	1	1.5.1
3.	MFCC uses (A). Filter banks and tan transform (B). Features and sine transform **(C). Filter banks and cosine transform** (D). Features and cosine transform	1	2	1	1.5.1
4.	How we can describe the state of the process in HMM? (A). Literal (B). Single random variable **(C). Single discrete random variable** (D). None of these	1	2	1	1.5.1
5.	Which of the following algorithm is applicable for solving temporal probabilistic reasoning? (A). Hill-climbing search algorithm **(B). Hidden Markov model** (C). Depth-first search algorithm (D). Breadth-first search algorithm	1	2	1	1.5.1

6.	Which function is used to load the letters data? **(A). crf.load_data()** (B). loaddata() (C). load() (D). data()	1	3	1	1.5.1
7.	What is the method used to train the CRF? (A). train() (B). crf() **(C). crf.train()** (D). crf_train()	1	3	1	1.6.1
8.	Which one is not a form of time series data? (A). int64 (B). float64 (C). bool **(D). Double**	1	3	1	1.5.1
9.	How to import numpy? **(A). import numpy as np** (B). import numpy c (C). import np (D). import numpy as num	1	3	1	1.6.1
10.	How to convert the data into a pandas data frame? (A). DataFrame() **(B). pd.DataFrame()** (C). pd_DataFrame() (D). pdDataFrame()	1	3	1	1.6.1

PART B (4 X 4= 16)
ANSWER ANY FOUR QUESTIONS

Q. No.	Questions	Marks	CO	BL	PI
11	**What is the method to avoid overfitting?** **Methods (4)** Early stopping. Pruning Regularization Ensembling Data augmentation.	4	2	1	1.3.1
12	**Differentiate supervised and unsupervised machine learning.** **Supervised (2)** **Unsupervised(2)**	4	2	1	2.1.1

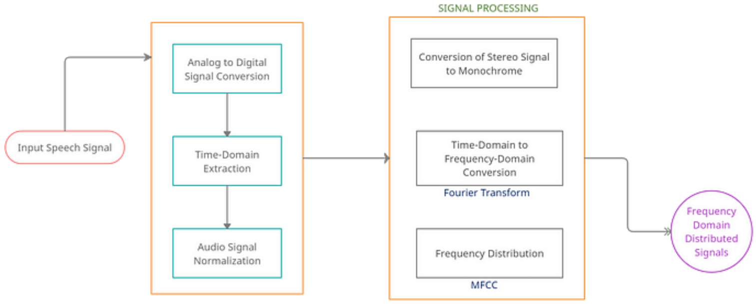
Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.

13	**Why instance-based learning algorithm sometimes referred to as Lazy learning algorithm?** In machine learning, lazy learning can be described as a method where induction and generalization processes are delayed until classification is performed. Because of the same property, an instance-based learning algorithm is sometimes called lazy learning algorithm.	4	2	1	1.1.2
14	**Illustrate about Conditional Random Fields.** **Conditional Random Fields** or **CRFs** are a type of probabilistic graph model that take neighboring sample context into account for tasks like classification. Prediction is modeled as a graphical model, which implements dependencies between the predictions. Graph choice depends on the application, for example linear chain CRFs are popular in natural language processing, whereas in image-based tasks, the graph would connect to neighboring locations in an image to enforce that they have similar predictions.	4	3	2	1.1.2
15	**How Hidden Markov Models performs in python?** **Steps (2)** Markov models are a useful class of models for sequential-type of data. 1. Initializing a Markov chain 2. Modelling transitions between states 3. Equilibrium or Stationary Distribution **Code(2)** `import numpy as np` `p_init = np.array([1/3., 1/3., 1/3.])` `p_init = np.array([0.1, 0.8, 0.1])` `p_transition = np.array(` `    [[0.90, 0.05, 0.05],` `     [0.01, 0.90, 0.09],` `     [0.07, 0.03, 0.9]]` `)` `p_transition`	4	3	2	1.3.1

	`p_state_t = [p_init]` `for i in range(200):  # 200 time steps sorta, kinda, approximates infinite time :)` `    p_state_t.append(p_state_t[-1]@ p_transition_example)` `state_distributions.plot();`				
16	**Write a program to display users selected year calendar on to the console.** import calendar year = int(input ("Please enter the Year: ")) # Here, it will take the year month = int(input ("Please enter the month: ")) # Here, it will take the month # Now, we will display the calendar Print ("The Calendar of: ", calendar.month(year, month))	4	3	3	2.1.1

PART C (2 X 12= 24)
ANSWER THE QUESTIONS

Q. No.	Questions	Marks	CO	BL	PI
17	**a) Explain the concept of transforming audio signals into the frequency domain** **Sampling (6)** **Quantization (6)** Speech is the primary form of human communication and is also a vital part of understanding behaviour and cognition. Speech Recognition in Artificial Intelligence is a technique deployed on computer programs that enables them in understanding spoken words. As images and videos, sound is also an analog signal that humans perceive through sensory organs. For machines, to consume this information, it needs to be stored as digital signals and analyzed through software. The conversion from analog to digital consists of the below two processes: **Sampling:** It is a procedure used to convert a time-varying (changing with time) signal s(t) to a discrete progression of real numbers x(n). Sampling period (Ts) is a term that defines the interval between two successive discrete samples. Sampling Frequency (fs = 1/Ts) is the inverse of the sampling period. Common sampling frequencies are 8 kHz, 16 kHz, and 44.1 kHz. A 1 Hz sampling rate means one sample per second and therefore high sampling rates mean better signal quality. **Quantization:** This is the process of replacing every real number generated by sampling with an approximation to obtain a finite precision (defined within a range of bits). In the majority of scenarios, 16 bits per sample are used for the representation of a single quantized sample. Therefore, raw audio samples generally have a signal range of -215 to 215 although, during analysis, these values are standardized to the range (-1, 1) for simpler validation and model training. A sample resolution is always measured in bits per sample.	12	2	2	2.5.2

	A general Speech Recognition system is designed to perform the tasks mentioned below and can easily be correlated with a standard data analytics architecture: The capture of speech (words, sentences, phrases) given by a human. You can think of this as the Data Acquisition part of any general Machine Learning workflow. Transforming audio frequencies to make it machine-ready. This process is the data pre-processing part where we clean features of the data for the machine to process it. Application of Natural Language Processing (NLP) on the acquired data to understand the content of speech. Synthesis of the recognized words to help the machine speak a similar dialect.  				
	OR				
	b) Describe the Hidden Markov Model for the application of Speech Recognition. **Steps (6)** **Expanation (6)** The first thing to do before classifying speech using HMM is to extract its feature and store it in a vector using MFCC. The MFCC process as follows: Frame the signal into short frames (20–40 ms) For each frame calculate the Discrete Fourier Transform Compute the mel-spaced filter-bank Take the logarithm of all filter-bank energies Take the Discrete Cosine Transform (DCT) of the log filter-bank energies. Keep 13 DCT coefficients and discard the rest. The next step after obtaining feature vectors is quantized the feature vectors using vector quantization. Vector quantization quantizes the continuous feature vectors to discrete feature vectors that can be processed by discrete HMM. A. Hidden Markov Model The HMM is a very powerful statistical method of characterizing the observed data samples of a discrete-time series [4]. This model describes all the possible paths through the state space and assigns a probability to each one. To understand the concepts of HMM, the following elements are defined as: [1]	12	2	2	2.6.2

N: No of states in HMM
M: Total different symbol per state
π: Initial state distribution (π=πi)
A: State transition probability distribution A=[aij]
B: Observation symbol probability distribution

π, A and B from [1] are called HMM model and notated by λ(λ=(A,B,π)). In order to apply HMM, there are 3 things that need to be done: [5]

1) Calculate Parameter
The main purpose of this step is to compute probability of observation sequence O={O1,O2,OT} given the model. The algorithm that used in this step is forward and backward algorithm.

2) Find Optimal State Sequence
The most common solution to find optimal state sequence is viterbi algorithm. Viterbi algorithm is a dynamic programming algorithm that calculates transition state path given observation sequence of symbols [6].

3) Estimating the Model Parameters
Estimation of model parameters is needed in order to adjust the model parameter (A,B,π), according to a certain optimally criteria. Baum-Welch algorithm is one of the techniques that used to solve this problem. It is an iterative method to estimate the new values for the model parameters.

B. Voice Activity Detection
Another important issue in speech recognition system is to determine active speech periods and silent periods within a given speech signal [7]. Speech can be characterized as discontinue signal because information is present only if someone is speaking. The regions where information presents called active region and the pauses between talking are called inactive or silence region. An algorithm employed to detect the presence or absence of speech is referred to as a voice activity detector (VAD) [7].

In general, VAD takes the feature from an input signal, spit those signals into frames (5-40ms), and then compare those value with threshold taken from the region that only contains noise. A sound is present (VAD = 1) if the value exceeds the threshold and not present or silent if the value is lower the threshold (VAD = 0). The success of VAD algorithm in splitting the speech depends on the threshold value.

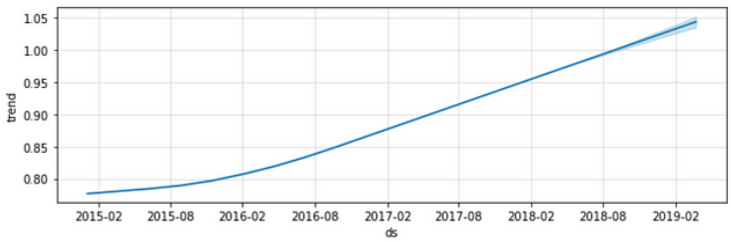
VAD algorithm that used in this system is an Energy-based VAD. A threshold value is calculated by averaging first 40 frames (10ms per frame) with the assumption that user will not speak in first 0.4 seconds after record button is clicked.

<table>
<tr><td></td><td>Assume that x(i) is ith speech sample. If frame length is N, then the jth frame can be written like (1).
fj={x((j−1)N+1)…x(j.N))}(1)
View SourceRight-click on figure for MathML and additional features.Energy value can be calculated by using (2)
Ej=1N∑i=(j−1)N+1j.N x2(i)(2)
View SourceRight-click on figure for MathML and additional features.where Ej is energy at jth frame and xi is speech sample at jth frame.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>18</td><td>

a) Describe the Time series and sequential data with examples.
Time Series Definition (2)
The components of time-series data (3)
Diagram(1)

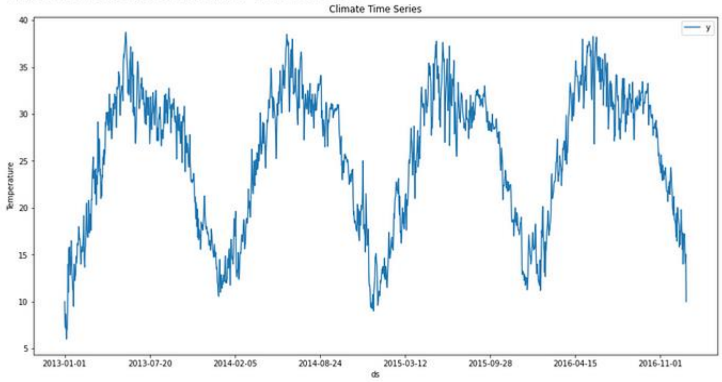
A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. In plain language, time-series data is a dataset that tracks a sample over time and is collected regularly. Examples are commodity price, stock price, house price over time, weather records, company sales data, and patient health metrics like ECG.

Trend — The data has a long-term movement in a series, whether it’s upwards or downwards. It may be caused by population growth, inflation, environmental change or the adoption of technology. Examples could be the long-term increase in the US stock market in the past ten years,



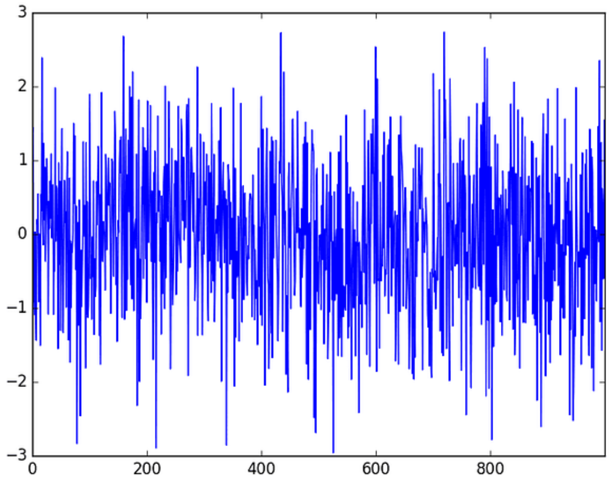
Seasonality — The data is correlated with calendar-related effects, whether it’s weekly, monthly, or seasonally, and it’s domain-specific. For example, for most e-commerce platforms, their sales around December rise because of Christmas.

`<matplotlib.axes._subplots.AxesSubplot at 0x7fe36fe92890>`



</td><td>12</td><td>3</td><td>3</td><td>2.5.2</td></tr>
</table>

White noise is the extreme situation of noise that has no trend and seasonality. Therefore it's nearly impossible to predict, and it's a kind of stationary time-series data.



Sequential Data Definition (2)
Examples (3)
Diagram(1)

Whenever the points in the dataset are dependent on the other points in the dataset the data is said to be Sequential data. A common example of this is a Timeseries such as a stock price or a sensor data where each point represents an observation at a certain point in time.

There are other examples of sequential data like sequences, gene sequences, and weather data.

timeseries	temperature	humidity
2019-11-01	30.49789813	68.43105695
2019-11-02	34.17009898	55.73184495
2019-11-03	38.72460807	57.63061612
2019-11-04	32.59569289	66.4557475
2019-11-05	30.04958949	59.94638509
2019-11-06	33.35118282	58.42986496
2019-11-07	32.42705438	68.35190681
2019-11-08	35.35606403	51.16645954
2019-11-09	35.97581918	54.28133089
2019-11-10	32.54127693	60.67027843
2019-11-11	30.32410699	51.22538266
2019-11-12	37.01971412	59.14352241
2019-11-13	37.64485094	58.7412217
2019-11-14	33.1106542	50.24747159

OR

b) Explain Analyzing stock market data using Hidden Markov Models. **Coding (9)** **Output (3)** Stock market data is a good example of time series data where the data is organized in the form of dates. In the dataset that we will use, we can see how the stock values of various companies fluctuate over time. Hidden Markov Models are generative models that are used to analyze such time series data.	12	3	2	2.6.2

1. Create a new Python file, and import the following packages:

```
import datetime
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.finance import
quotes_historical_yahoo_ochl
from hmmlearn.hmm import GaussianHMM
```

2. Get the stock quotes from Yahoo finance. There is a method available in matplotlib to load this directly:

```
# Get quotes from Yahoo finance
quotes = quotes_historical_yahoo_ochl("INTC",
        datetime.date(1994, 4, 5),
datetime.date(2015, 7, 3))
```

3. There are six values in each quote. Let's extract the relevant data such as the closing value of the stock and the volume of stock that is traded along with their corresponding dates:

```
# Extract the required values
dates = np.array([quote[0] for quote in quotes],
dtype=np.int)
closing_values = np.array([quote[2] for quote in
quotes])
volume_of_shares = np.array([quote[5] for quote
in quotes])[1:]
```

4. Let's compute the percentage change in the closing value of each type of data. We will use this as one of the features:

```
# Take diff of closing values and computing rate
of change
diff_percentage = 100.0 *
np.diff(closing_values) / closing_values[:-1]
dates = dates[1:]
```

5. Stack the two arrays column-wise for training

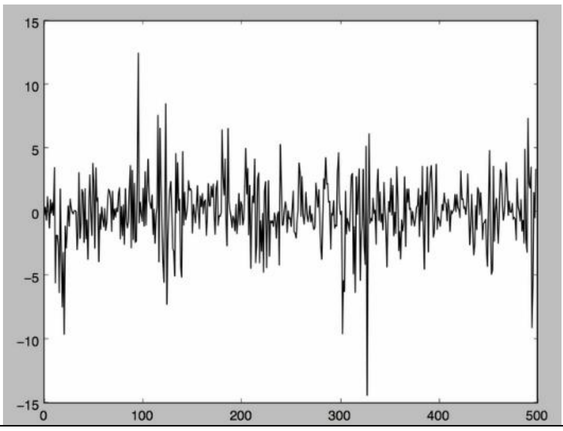
```
# Stack the percentage diff and volume values
column-wise for training
X = np.column_stack([diff_percentage,
volume_of_shares])
```

6. Train the HMM using five components:

```
# Create and train Gaussian HMM
print "\nTraining HMM...."
model = GaussianHMM(n_components=5,
covariance_type="diag", n_iter=1000)
model.fit(X)
```

7. Generate 500 samples using the trained HMM and plot this, as follows:

```
# Generate data using model
num_samples = 500
samples, _ = model.sample(num_samples)
plt.plot(np.arange(num_samples), samples[:,0],
c='black')
plt.show()
```



COURSE COORDINATOR **HOD/CSE**