University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 1

Programming in PROLOG

1. Write a PROLOG program for addition and multiplication of two numbers.
2. Write a PROLOG program to find the sum of all the numbers in a list.
3. Write a PROLOG program to compare two strings taken as input.
4. Write a PROLOG program to determine if an element is a member of a list or not.
5. Write a PROLOG program to reverse a list.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 2

Programming in PROLOG

1. Write a PROLOG program to determine all the permutations of all the elements of a given list.
2. Write a PROLOG program that stores information about your family and can answer queries about the relationships.
3. Write a PROLOG program to find the factorial of a number.
4. Write a PROLOG program to check if a number is prime or not.
5. Write a PROLOG program to print the fibonacci series upto n terms, where n is given as input.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 3

Introduction to Numpy

1. Make the array [0, 1, 2] with np.array
2. Make the array [0, 1, 2] with np.arange
3. Make the array [0, 0, 0] with np.zeros
4. Make the matrix [[0, 1], [2, 3]] with np.array
5. Make the matrix [[0, 1], [2, 3]] with np.arange and np.reshape.
6. Print the shape of the matrix from the previous part with .shape.
7. Print the number of rows in the matrix from part 5 with .shape.
8. Note that the number of rows is the first element of the shape tuple.
9. Take the average of elements in a vector (np.average).
10. Add two vectors element-wise, which means that the each element in the result vector is the sum of the corresponding elements in two vectors.
11. Take the sum of elements in a vector (np.sum).
12. Take the square root of a number (np.sqrt)

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 4

Introduction to Pandas

1. Use pandas to create a student mark-sheet with Student ID, Name, and marks obtained in 6 subjects
2. Use pandas to get the name and ID of the student with the highest percentage of marks.
3. Use pandas to identify the student who got lowest marks in more than 2 subjects.
4. Save the dataFrame as a csv file called marksheet.csv.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 5

Linear Regression – Supervised Learning (Regression)

1. a. Load the boston dataset from sklearn.datasets and display its description.
b. Keep the data in a dataframe.
c. Find the self-correlation of the data features, upto 2 decimal places.
d. Add boston.target as the target variable "MEDV" to the dataframe, for regression
e. Determine the variables which have minimum and maximum self-correlation values and use them as training variables (x).
f. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets
g. With the use of sklearn.linear_model module, load the Linear Regression model, train it using training data (x_train) and get the predictions on test data (y_pred)
h. Determine and print the Mean Squared Error, Root Mean Squared Error, and R2 score for the regression model performance

2. Keeping everything same till Q1 part f, train a Ridge regression model with same split ratio, feature variables and target. For the Ridge regression, use alpha values in a range from 0 upto 100 with a step size of 5, i.e. $0<=\alpha <=100$, $\alpha=\alpha+5$. Record these 21 values of RMSE and R2 scores in separate lists.

3. Keeping everything same till Q1 part f, train a Lasso regression model with same split ratio, feature variables and target. For the Lasso regression, use alpha values in a range from 0 upto 100 with a step size of 5, i.e. $0<=\alpha <=100$, $\alpha=\alpha+5$. Record these 21 values of RMSE and R2 scores in separate lists.

4. Using the RMSE and R2 score lists from Q2 and Q3, plot the values in two separate line plots with the help of matplotlib. The first plot should illustrate the RMSE scores achieved using Ridge and Lasso regression for given set of α values, and the second plot should illustrate the R2 scores for the same.

5. Keeping everything same till Q1 part f, train a ElasticNet regression model with same split ratio, feature variables and target. For the ElasticNet regression, use l1_ratio values in a range

from 0 upto 1 with a step size of 0.1, i.e. 0<=l1_ratio <=1, l1_ratio=l1_ratio+0.1 Record the corresponding 10 values of RMSE and R2 scores in separate lists. Plot these RMSE and R2 score values in separate plots.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 6

Logistic Regression– Supervised Learning (Classification)

1. a. Load the digits dataset from sklearn.datasets and display its description.
b. Keep the data in a dataframe.
c. Find the shapes of digits.data and digits,target.
d. Choose the n-th image pixel data, where n = last 2 digits of your roll number. Determine the class of the n-th image.
e. Reshape the chosen image pixel data to a 8x8 numpy array
f. Using matplotlib, plot the 8x8 numpy array with 'binary' colormap.
g. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y
g. With the use of sklearn.linear_model module, load the Logistic Regression model, train it using training data (x_train) and get the predictions on test data (y_pred). Use 3 different models with max_iter values as 100, 500, 1000, 5000 and 10000
h. Determine and record all the R2 scores in a list.

2. Repeat the experiment as given in Q1, using split ratios of 60:40 and 80:20. Record the R2 scores in separate lists.

3. Plot the R2 scores from the 3 lists obtained from Q1 and Q2 and illustrate how the R2 scores vary for different split ratios and different max_iter values.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 7

Decision Tree and Random Forest Regression – Supervised Learning (Regression)

1. a. Load the boston dataset from sklearn.datasets and display its description.
b. Keep the data in a dataframe.
c. Find the self-correlation of the data features, upto 3 decimal places.
d. Add boston.target as the target variable "MEDV" to the dataframe, for regression
e. Determine the variables which have minimum and maximum self-correlation values and use them as training variables (x).
f. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets
g. With the use of sklearn.tree module, load the DecisionTreeRegressor model, train it using training data (x_train) and get the predictions on test data (y_pred)
h. Determine and record the Mean Squared Error, Root Mean Squared Error, and R2 score for the regression model performance, in separate lists.

2. Repeat the experiment as given in Q1, using split ratios of 60:40 and 80:20. Record the Mean Squared Error, Root Mean Squared Error, and R2 scores in separate lists for each case.

3. Keeping everything same till Q1 part f, train a RandomForestRegressor model from sklearn.ensemble module. Use the different sets of split ratios as mentioned in Q2. Record the Mean Squared Error, Root Mean Squared Error, and R2 scores in separate lists for each case.

4. Plot the Mean Squared Error, Root Mean Squared Error, and R2 scores in separate plots using matplotlib module. For each plot, use the calculated scores for the 3 different splits of train and test data. Do this for both the DecisionTreeRegressor and RandomForestRegressor models.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 8

Decision Tree and Random Forest Classification– Supervised Learning (Classification)

1. a. Load the digits dataset from sklearn.datasets and display its description.
b. Keep the data in a dataframe.
c. Find the shapes of digits.data and digits,target.
d. Choose the n-th image pixel data, where n = last 2 digits of your roll number *2. Determine the class of the n-th image.
e. Reshape the chosen image pixel data to a 8x8 numpy array
f. Using matplotlib, plot the 8x8 numpy array with 'binary' colormap.
g. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y
g. With the use of sklearn.tree module, load the DecisionTreeClassifier model, train it using training data (x_train) and get the predictions on test data (y_pred). Use 3 different models with max_depth values as 10, 20, 30, 40 and 50
h. Determine and record all the R2 scores in a list.

2. Repeat the experiment as given in Q1, using split ratios of 60:40 and 80:20. Record the R2 scores in separate lists.

3. Plot the R2 scores from the 3 lists obtained from Q1 and Q2 and illustrate how the R2 scores vary for different split ratios and different max_depth values.

4. Repeat the above set of experiments for the RandomForestClassifier from sklearn.ensemble.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 9

K-Means Clustering – Un-supervised Learning

1.a. From sklearn.datasets module, use make_blobs to generate data-set of 200 samples with 3 centers, 2 features, cluster standard-deviation of 1.5 and random state = 30
b. Plot the generated data as a scatter plot using matplotlib
c. Use the kMeans clustering algorithm for finding 3 clusters from the generated data
d. Determine and record the cluster predictions and cluster centres
e. Plot the clusters with respective cluster centres (use different colors for each cluster and black for the cluster centres)

2. a. Load the iris dataset from sklearn.datasets and display its description.
b. Keep the data in a dataframe.
c. Select the features sepal_length, sepal_width, petal_length and petal_width for clustering
d. Plot the selected data as a scatter plot using matplotlib
e. Use the kMeans clustering algorithm for finding 3 clusters from the generated data
f. Determine and record the cluster predictions and cluster centres
g. Plot the clusters with respective cluster centres (use different colors for each cluster and black for the cluster centres)

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 10

Convolutional Neural Network – Supervised Learning (Classification)

1.a. Load the digits dataset from sklearn.datasets module
b. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y
c. Reshape the train and test data to represent data points as 8x8 pixels and in grayscale form (represented by 1)
d. Using keras.utils, convert the train and test labels to categorical values (use one-hot encoding)
e. Create a 2D convolutional neural network with the following layers in order:
i. A 2D convolutional layer with 8 filters and 3 layers that takes as input 8x8 pixels in grayscale form (represented by 1),
ii. A Max-pooling layer with pool size of 2,
iii. A flatten layer ,
iv. A Dense layer with 10 neurons that uses the softmax activation function,
f. Compile the model from question 1e using adam optimizer, categorical cross-entropy loss function and accuracy as the evaluation metric.
g. Train the model and use test data for validation, using 10 epochs and record the accuracy score.

2. Repeat the experiment in Q1 with different split ratios as 60:40 and 80:20. Record the accuracies in each case. Plot the accuracy values as recorded in Q1 and Q2 for different split ratios.

3. Using the models with separate split ratios as defined in Q1 and Q2, repeat the experiments by increasing the number of epochs to 20 and 30. Record the new values of accuracy and plot them for each epoch value.

4. Using the models with separate split ratios as defined in Q1 and Q2, repeat the experiments by increasing the number of convolutional layer filters to 16 and 32, keeping the number of epochs fixed at 10. Record the new values of accuracy and plot them for each convolutional filter size.

5.a. Load the MNIST dataset from keras.datasets module and create the training (x_train, y_train) and test (x_test, y_test) datasets

b. Plot the n-th image pixel data, where n=3*(last 2 digits of your roll number).

c. Reshape the train and test data to represent data points as 28x28 pixels and in grayscale form (represented by 1)

d. Using keras.utils, convert the train and test labels to categorical values (use one-hot encoding)

e. Create a 2D convolutional neural network with the following layers in order:

   i. A 2D convolutional layer with 8 filters and 3 layers that takes as input 28x28 pixels in grayscale form (represented by 1),

   ii. A Max-pooling layer with pool size of 2,

   iii. A flatten layer ,

   iv. A Dense layer with 10 neurons that uses the softmax activation function,

f. Compile the model from question 1e using adam optimizer, categorical cross-entropy loss function and accuracy as the evaluation metric.

g. Train the model and use test data for validation, using 10 epochs and record the accuracy score.

6. Using the model from experiment in Q5, repeat the experiment by increasing the number of epochs to 20 and 30. Record the new values of accuracy and plot them for each epoch value.

7. Using the model as defined in Q5, repeat the experiment by changing the number of convolutional layer filters to 16 and 32. keeping the number of epochs fixed at 10. Record the new values of accuracy and plot them for each convolutional filter size.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 11

Recurrent Neural Network – Supervised Learning (Classification)

1.a. Load the MNIST dataset from keras.datasets module and create the training (x_train, y_train) and test (x_test, y_test) datasets

b. Plot the n-th image pixel data, where n=5*(last 2 digits of your roll number).

c. Normalize the image pixel data (both train and test data) to values in the range of 0 to 255 after they are converted to float

d. Using keras.utils, convert the train and test labels to categorical values (use one-hot encoding)

e. Create a 2D LSTM neural network with the following layers in order:

i. A LSTM layer with 128 units that takes the training data as input and uses return sequences.

ii. A dropout layer where the dropout rate is set to 20%,

iii. Another LSTM layer with 128 units,

iv. A Dense layer with 64 neurons that uses relu as the activation function,

v. Another dropout layer where the dropout rate is set to 20%,

vi. A Dense layer with 10 neurons that uses softmax as the activation function,

f. Compile the model from question 1e using adam optimizer, categorical cross-entropy loss function and accuracy as the evaluation metric.

g. Train the model and use test data for validation, using 5 epochs and record the accuracy score.

2. Using the model from experiment in Q1, repeat the experiment by increasing the number of epochs to 10, 15 and 20. Record the new values of accuracy and plot them for each epoch value.

3. Using the model as defined in Q1, repeat the experiment by changing the number of LSTM layer units to 64 and 256, keeping the number of epochs fixed at 10. Record the new values of accuracy and plot them for each LSTM unit size.

4.a. Load the digits dataset from sklearn.datasets module

b. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y

c. Plot the n-th image pixel data, where n=7*(last 2 digits of your roll number).
d. Normalize the image pixel data (both train and test data) to values in the range of 0 to 255 after they are converted to float
e. Using keras.utils, convert the train and test labels to categorical values (use one-hot encoding)
f. Create a 2D LSTM neural network with the following layers in order:

i. A LSTM layer with 128 units that takes the training data as input and uses return sequences,

ii. A dropout layer where the dropout rate is set to 20%,

iii. Another LSTM layer with 128 units,

iv. A Dense layer with 64 neurons that uses relu as the activation function,

v. Another dropout layer where the dropout rate is set to 20%,

vi. A Dense layer with 10 neurons that uses softmax as the activation function,

g. Compile the model from question 1f using adam optimizer, categorical cross-entropy loss function and accuracy as the evaluation metric.
h. Train the model and use test data for validation, using 5 epochs and record the accuracy score.

5. Repeat the experiment in Q4 with different split ratios as 60:40 and 80:20. Record the accuracies in each case. Plot the accuracy values as recorded in Q4 and Q5 for different split ratios.

6. Using the models with separate split ratios as defined in Q4 and Q5, repeat the experiments by increasing the number of epochs to 10, 15 and 20. Record the new values of accuracy and plot them for each epoch value.

7. Using the models with separate split ratios as defined in Q4 and Q5, repeat the experiments by changing the number of LSTM layer units to 64 and 256. Keep the epochs fixed at 10. Record the new values of accuracy and plot them for each LSTM unit size.

University of Engineering & Management, Kolkata

Department of Computer Science and Engineering

Paper Name: Artificial Intelligence & Machine Learning Laboratory

Paper Code: PCC-CSE495

List of Experiments - Week 12

Label Propagation and Label Spreading – Semi-supervised Learning

1. a. Load the digits dataset from sklearn.datasets module
b. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y
c. Retrieve the indices for all the data in training set
d. Using the indices from question c, keep the first 300 training labels intact, and set all remaining training data labels to -1
e. Using Label Propagation from sklearn.semi_supervised (without setting gamma), fit the training data
f. Determine the R2 score
g. Use Label Propagation on the same data from question e, but set gamma to 0.3
h. Determine the new R2 score
i. Retrieve the labels as determined by Label Propagation algorithm
j. Find the confusion matrix using original training labels and labels determined by the algorithm (consider indices from 300 onwards)

2.a. Load the digits dataset from sklearn.datasets module
b. Using a 70:30 split ratio, create the training (x_train, y_train) and test (x_test, y_test) datasets, where digits.data is x and digits.target is y
c. Retrieve the indices for all the data in training set
d. Using the indices from question iii, keep the first 300 training labels intact, and set all remaining training data labels to -1
e. Using Label Spreading from sklearn.semi_supervised (without setting gamma), fit the training data
f. Determine the R2 score
g. Use Label Spreading on the same data from question iv, but set gamma to 0.5
h. Determine the new R2 score
i. Retrieve the labels as determined by Label Spreading algorithm
j. Find the confusion matrix using original training labels and labels determined by the algorithm (consider indices from 300 onwards)

3. Repeat the experiments in Q1 and Q2 for different values of gamma in the range of 0.1 to 1 with step size of 0.1. Determine the accuracy of the model in each case. Plot the change in accuracy values with respect to value of gamma for both models.