| **MACHINE LEARNING** | |
| --- | --- |
| **Course Code: ISL66** | **Credits: 0:0:1** |
| **Prerequisites: Scripting Languages** | **Contact Hours: 14P** |
| **Course Coordinator: Dr.Mydhili K Nair** |  |

**Laboratory Experiments:**

| **Implement the following programs using Python** | |
| --- | --- |
| 1. **Supervised Learning Algorithms - Linear Regression:** 2. **Simple Linear Regression - Univariate** 3. **Multiple Linear Regression - Multivariate**   Consider ***any dataset*** from UCI repository. Create Simple and Multiple Linear Regression models using the training data set. Predict the scores on the test data and find the error in prediction (E.g. RMSE). Include appropriate code snippets to visualize the model. Use Sub-Plots Interpret the result. Write the Inference.   1. **Model Measurement Analysis**: Using ***any dataset*** and ***any classifier*** do the following: 2. Calculate TP, TN, FP and FN from sklearn library functions 3. Calculate different metrics (Accuracy, Precision, Recall(Sensitivity), F1-Score, MCC, Specificity, Negative Predictive Value) by defining our own functions 4. Get the sklearn metrics of these values 5. Verify them by comparing with scikit-learn's library functions. 6. Get the result of Confusion Matrix using sklearn 7. Using sklearn, plot the ROC Curve of the probability values in our test data 8. Using sklearn, plot the ROC Curve of random probabilities 9. Calculate the AUC of our test data using sklearn 10. Calculate the AUC of random probabilities using sklearn 11. Interpret the results. Write the inference/analysis of each output. 12. **Probabilistic Supervised Learning - Naive Bayes:** Create a dataset from the sample given to you(e.g. “Play Tennis Probability”, “Shopper Buying Probability” etc.). Perform the necessary pre-processing steps such as encoding. Train the model using Naive Bayes Classifier. Give new test data and predict the classification output. Handcode the classification probability and compare with the model output. Analyze and write the inference. 13. **Artificial Neural Networks - Multi Layer Perceptron:** Write a program to construct a single layer Perceptron for a dataset     1. Generate a random dataset as a file. The dataset should have at least 100 records. Each record should have at least 4 floating point features and a binary label (0 - negative or 1 - positive)     2. The program should contain functions to        1. Read the dataset from the file.        2. Split the data into train and test. Ensure the data is split in the same way every time the program runs.        3. Initialize the weights of the perceptron, learning rate and epochs        4. Define the activation function        5. Train the model i.e. Learn the weights of the perceptron on the training data.        6. Print the learned weights and the hyperparameters (epoch and learning rate)        7. Predict the outputs on train and test data        8. Print the confusion matrix, accuracy, precision, recall on train and test data 14. **Supervised Learning Algorithms - Decision Trees:** Implement decision trees considering a data set of your choice. 15. Create an ID3 Decision Tree. Write the inference 16. Hardcode the Entropy and Information Gain calculation.   **(OR)**  Handcode the Gini Index Calculation.   1. Do not use a pre-defined sklearn library for (b) or (c ) above. 2. Separate out the predicting and predictor attributes. 3. Print the values of Entropy and Information Gain at each step.   (OR)  Print the values of Gini Index at each step.   1. Be verbose and print out the count of occurrence of predictor attributes at each step. 2. **Supervised Learning Algorithms - Logistic Regression:** Implement logistic regression and test it using any dataset of your choice from UCI repository. The output should include Confusion Matrix, Accuracy, Error rate, Precision, Recall and F-Measure. 3. **Supervised Learning Algorithms - KNN:** Build a KNN model for prediction whether a person will have diabetes or not with a high accuracy score: 4. Perform some appropriate Pre-Processing steps on the given dataset for better results 5. Implement the KNN algorithm on your own. (Don’t use any pre built code/lib) 6. Try other possible processes that can be done to dataset and tuning the model to increase   accuracy.   * Increase K value * Normalization * Different Distance Metrics  1. Perform Feature Ablation Study 2. Additional Tries: Weight the features before doing KNN prediction. 3. **Un-Supervised Learning Algorithms - K-Means Clustering:** Build a K-Means Model for the given dataset. So in K-Means choosing the K value that gives a better model is always a challenge, As we increase value of K with dataset having n points, the likelihood of the model increases, and obviously K<N, so rank or maximize the likelihood we use BIC(read about Bayesian Information Criterion for better understanding, before attempting the question).Now,    1. Build a K-Means Model for the given Dataset (You can use the library funct.)    2. Implement the BIC function that takes the cluster and data points and returns BIC value    3. Implement a function to pick the best K value, that is maximize the BIC.    4. Visualize the pattern found by plotting K v/s BIC. 4. **Un-Supervised Learning Algorithms - Hierarchical Clustering:** Using any dataset from the UCI repository implement ***any one*** type of Hierarchical Clustering you are familiar with.    1. Plot the Dendrogram for Hierarchical Clustering and analyze your result.    2. Plot the clustering output for the same dataset using these two partitioning techniques.    3. Compare the results.    4. Write the inference. 5. **Supervised Learning Algorithms - Support Vector Machines:** Use SVM to classify the flowers in Iris dataset. Visualize the results for each of the following combinations:   **(a)** For every pair of (different) features in the dataset (there are 4). Which pair separates the data easily?  **(b)** Using One-vs-Rest and using One-vs-One. Which one fits better? Which one is easier to compute? Why?  **(c)** Using different kernels (Linear, RBF, Quadratic). | |
| **Reference:**   1. 1. Stephen Marsland, “Machine Learning - An Algorithmic Perspective”,Second Edition, CRC Press - Taylor and Francis Group, 2015 2. 2. Ethem Alpaydin, “Introduction to Machine Learning”, Second Edition, MIT Press, Prentice Hall of India (PHI) Learning Pvt. Ltd. 2010 | |
| **Course Outcomes (COs):**  At the end of the course, student will be able to - | |
| **1.** | Design and implement the various Machine Learning Algorithms in the realm of supervised and unsupervised learning. **(PO – 1(2),2(3),3(3),4(3),5(2),12(3)) & (PSO – 1(3), 2(2))** |
| **2.** | Demonstrate the working principle of these different ML models, determine their performance, usage and their applications. **(PO – 1(2), 3(2), 10(3)) & (PSO – 1(3),3(3))** |
| **3.** | Analyze the results and produce substantial written documentation. **(PO – 1(2),4(3),10(3),12(2)) & PSO – (1(2),2(2),3(2))** |

**Conduction of Practical Examination:- (50 Marks)**

* All laboratory experiments are to be included for practical examination.
* Marks Distribution:
* Procedure Writing (20 Marks)
* Implementation and Testing (20 Marks)
* Viva (10 Marks)