UNIT – III: Calculus of Variations (15 hours)

Variational problems, Variation of a functional and its properties, Extremum of functional, Necessary condition for an extremum, Euler's equation and its generalization, Variational derivative, General variation of a functional and variable end point problem, Sufficient conditions for the extremum of a functional.

Essential Readings

1. Gelfand, I. M. and Fomin, S.V. (2000). Calculus of Variations. Dover Publications, Inc.
2. Krasnov, M., Kiselev, A. and Makarenko, G. (1971). Problems and Exercises Integral Equations, Mir Publication Moscow.
3. Logan, J. David (1987). Applied Mathematics: A Contemporary Approach, John Wiley & Sons, Inc.

Suggestive Readings

- Hildebrand, F. B. (1992). Methods of Applied Mathematics (2nd ed.). Dover Publications.
- Zemyan, Stephen M. (2012). The Classical Theory of Integral Equations: A Concise Treatment. Birkhäuser.

DISCIPLINE SPECIFIC ELECTIVE COURSE – 6(vi): MACHINE LEARNING: A MATHEMATICAL APPROACH

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Machine Learning: A Mathematical Approach	4	3	0	1	Class XII pass with Mathematics	Basic Knowledge of Python

Learning Objectives: The main objective of this course is to:

- Gain mathematical insights into the functioning of popular methods of Regression, Classification, Clustering and Dimension reduction.
- Understand the mathematical framework of learning and apply it to assess the performance of a number of regression, classification and density estimation algorithms
- Detect overfitting and employ regularization techniques to control it.

Learning Outcomes: This course will enable the students to:

- Learn how to build popular models of regression and classification including Linear regression, Polynomial regression, Logistic classifier, Support vector machine, Decision Tree, Random forests, Naïve Bayes classifier.
- Evaluate the performance of models on test data through analytical techniques (VC bounds and dimension) and Cross-validation to facilitate model selection and feature selection.
- Improve model performance by controlling overfitting through regularization techniques like Ridge and Lasso.

- Understand when to apply dimension reduction and combine it with other supervised learning methods.
- Understand and implement the key principles of Artificial Neural Networks in the context of regression and classification and employ them in function approximation.

SYLLABUS OF DSE-6(vi)

UNIT – I: Introduction to Machine Learning and its Applications (18 hours)

Overview of different tasks: Regression, Classification, and Clustering. Evaluation metrics– Mean Absolute error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE). Linear Regression, Cost function, Polynomial Regression, Gradient Descent Algorithm (GDA). Logistic Regression: Evaluation metrics - accuracy, precision, recall, confusion matrix, Receiver Operating Characteristic Curve (ROC curve) and Area Under ROC Curve (AUC), Vapnik-Chervonenkis (VC) dimension, VC bounds (only statement). *k*-fold validation, Concepts of training set, validation set and test set, Underfitting-Overfitting, Regularization techniques– Ridge, Lasso for Linear Regression and Logistic Regression, Bias-variance tradeoff.

UNIT – II: Popular Machine Learning Techniques (18 hours)

Cross-entropy and Gini Index, Decision Tree, Regression Tree, Random Forest and Bagging, Tree Pruning. Support Vector Machine (SVM), Kernel SVM (Gaussian) Similarity Criterion, *k*-Means clustering technique. Naive Bayes classifier- Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA). Dimensionality Reduction, Feature Selection, Principal Component Analysis (PCA).

UNIT – III: Introduction to Deep Learning (9 hours)

Artificial neural network (ANN), Activation functions – definition and examples (Sigmoid, ReLU, Tanh), neurons, layers, Cost function, Information passing, Back propagation algorithm, Optimizers, Learning rate, Statement of Universal Approximation Theorem for continuous functions, Regularization with ANN, Normalization.

Essential Readings

1. Abu-Mostafa, Y. S., Magdon-Ismail, M. & Lin, H.-T. (2012), Learning from Data, AML Book.
2. James, Gareth., Witten, D., Hastie, T., Tibshirani, R. and Taylor, J. (2023), An Introduction to Statistical Learning: with Applications in Python, Springer Nature Switzerland.
3. Ovidiu Calin, Springer. (2020). Deep Learning Architectures: A Mathematical Approach, Springer Nature Switzerland.

Suggestive Readings

- Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020), Mathematics for Machine Learning, Cambridge University Press.
- Shalev-Shwartz, S., and Ben-David, S. (2014), Understanding Machine Learning - From Theory to Algorithms, Cambridge University Press.
- Phillips, Jeff. (2020), Mathematical Foundations for Data analysis, Springer.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016), Deep Learning, MIT Press.
- Hastie, T.; Tibshirani, R., and Friedman, J. (2001), The Elements of Statistical Learning, Springer New York Inc.

Practical (30 hours)- Practical work to be performed in computer lab using Python.

Following lab exercises should be done for at least two classification problems or two regression problems or both, whenever applicable.

Following datasets can be used for classification problems

- https://scikit-learn.org/stable/datasets/toy_dataset.html
  (Toy datasets, Iris plants dataset, handwritten digits dataset, Wine recognition dataset, Breast cancer diagnostic dataset)
- https://pypi.org/project/ISLP/
  (Smarket dataset)

Following datasets can be used for regression problems

- https://scikit-learn.org/0.15/modules/classes.html#module-sklearn.datasets
  (Diabetes dataset, Boston house dataset, California housing dataset, Advertising dataset)

Following tasks needs to be performed for the below mentioned ML techniques in scikit learn (https://scikit-learn.org/stable/), whenever applicable:

- Split the dataset into two parts: training and test. Create and train model on training set and report model performance on test set.
- Test the model performance using k-fold cross validation (take k = 5 or 10) in terms of applicable metrics like – Accuracy, Precision, Recall, MAE, RMSE etc.
- Finding optimal parameters using Grid Search CV, whenever applicable; for example: in case of polynomial regression, employ Grid search CV to find the optimal value of the degree d for which the MSE is least.

<u>Practicals List</u>:

1. Create a Linear regression model. Use one variable at a time, all variable at a time, and statistically significant variables (using co-relation matrix) at a time, and observe the model performance. Preferably work with advertising dataset to predict sales in terms of the above features.
2. Fix a 10th order polynomial and sample 15 noisy data points (that is all 15 points do not lie on this polynomial). This is usually done by adding a white noise $\epsilon \sim N(0,1)$ to the polynomial $f(x)$. Using polynomial regression fit two models: one of order 10 and one of order 2. Compare the in-sample and out-sample errors for both models. Try to observe underfitting-overfitting, if any. In another scenario, take $f(x)$ to be a polynomial of order 50 and sample 15 noiseless data points (all lie on the graph of $f(x)$) and again fit a polynomial model of order 2 and 10. Compare the in-sample and out-sample errors. (refer to Exercise 4.2 and Problem 4.4 of [1]).
3. On the Smarket data, predict direction based on features Lag1 and Lag2. Split the Smarket dataset into training and testing parts in the ratio 80-20. Fit logistic regression on the training data and evaluate its accuracy on the test data via confusion matrix, ROC, and AUC. Plot decision boundary for the logistic regression in the 2D feature space spanned by Lag1 and Lag2 (you might need to rescale the variables).
4. Create decision tree models for classification and regression. Observe the effect of various parameters like - splitting criterion (Gini index, Cross entropy), max depth (for tree pruning). Examine overfitting-underfitting in the associated tree model. Display a decision tree.
5. Create Random Forest models for classification and regression. Observe the effect of number of estimators in the context of overfitting.

6. Create SVM models for classification and regression. Observe the effect of the parameter - kernel.
7. Create LDA and QDA models and assess them preferably on the digits dataset.
8. Create *k*-means cluster model for clustering. Observe the effect of parameter *k* (number of clusters). Plot *k* versus error to find out best *k* (Elbow criterion). Plot clusters in case of 2-dimensional data.
9. Demonstrate Principal Component Analysis (PCA) on a dataset with large number of features.
10. Create an ANN model for both classification and regression. Observe the effect of parameters-hidden layer sizes, activation functions (ReLU, Logistic/Sigmoid, Tanh), optimizers (Adam, Sgd), batch size, learning rate, early stopping, validation fraction, maximum number of iterations. Plot iteration number versus accuracy on training and validation dataset. The mnist dataset may be used to explore real strength of ANN. (https://www.kaggle.com/datasets/oddrationale/mnist-in-csv?resource=download in csv format).