III B.Tech II Semester

		L	T	P	C
23A39604	**ML MODEL OPTIMIZATION LAB (Professional Core)**	**0**	**0**	**3**	**1.5**

Course Objectives:

- To provide practical exposure to model optimization techniques for improving performance and efficiency of machine learning models.
- To explore various hyperparameter tuning methods and optimization algorithms.
- To apply regularization techniques to control overfitting and improve generalization.
- To evaluate model performance through various metrics and validation strategies.
- To learn pruning, quantization, and deployment strategies for optimized models in real-world applications.

Course Outcomes:

- Students will gain hands-on experience in optimizing machine learning models using advanced techniques like hyperparameter tuning, regularization, pruning, and quantization.
- Student will be able to assess model performance critically using evaluation metrics and implement strategies for reducing computational complexity.
- Students with the skills necessary to build efficient, scalable, and high-performing ML systems suitable for deployment in constrained environments such as edge devices and mobile platforms.

List of Lab Experiments:

1. Implementing Grid Search and Random Search
    - Hyperparameter tuning using Scikit-learn's GridSearchCV and RandomizedSearchCV
      (Cognitive Level: Apply & Analyze)
2. Bayesian Optimization using Hyperopt / Optuna
    - Optimize model hyperparameters using probabilistic approaches
      (Cognitive Level: Analyze & Evaluate)
3. Early Stopping in Training Deep Learning Models
    - Implement early stopping to avoid overfitting with TensorFlow/Keras
      (Cognitive Level: Apply & Evaluate)
4. Regularization Techniques (L1, L2, Dropout)
    - Apply different regularization methods to improve model generalization
      (Cognitive Level: Analyze)
5. Cross-Validation Techniques
    - K-Fold, Stratified K-Fold, and Leave-One-Out Cross-Validation
      (Cognitive Level: Apply & Evaluate)
6. Model Performance Evaluation
    - Use confusion matrix, precision, recall, F1-score, AUC-ROC for performance
      (Cognitive Level: Evaluate)
7. Model Pruning
    - Prune unimportant weights in a trained neural network using PyTorch/TensorFlow
      (Cognitive Level: Create & Evaluate)
8. Quantization Aware Training (QAT)
    - Train a quantized model to reduce memory usage without losing accuracy
      (Cognitive Level: Apply & Create)

9. Knowledge Distillation
    - Transfer knowledge from a large model to a small one (student-teacher model) (Cognitive Level: Analyze & Create)
10. Optimizing Model Inference Time

- Use ONNX, TensorRT, or OpenVINO for faster model inference (Cognitive Level: Apply & Evaluate)

11. Automated Machine Learning (AutoML)

- Use Auto-Sklearn, Google AutoML, or TPOT for full pipeline optimization (Cognitive Level: Evaluate & Create)

12. Model Deployment Optimization

- Optimize model for deployment using TFLite/ONNX in edge devices (Cognitive Level: Create & Evaluate)

Text Books:

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly.
2. Sebastian Raschka, Python Machine Learning, Packt Publishing.
3. Francois Chollet, Deep Learning with Python, Manning.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning.
3. Vijay Madisetti, Machine Learning and Optimization Models for Real-Time Applications, Springer.

Online Courses:

1. Model Optimization Techniques – Coursera (DeepLearning.AI)
2. Hyperparameter Tuning in Python – Udacity