College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 401 **TITLE:** Introduction to Machine Learning
   Units: 4.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; 48.0-54.0 Lab hours; 96.0-108.0 Homework hours; 192.0-216.0 Total Student Learning hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Prerequisite: CIS 400

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC
   AA/AS Degree Requirements:
   CSM - GENERAL EDUCATION REQUIREMENTS: E2b. Communication and Analytical Thinking

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   This course covers critical machine learning algorithms in AI, including classification (perceptrons, SVMs, Gaussian discriminant analysis), regression techniques (linear, logistic, polynomial, ridge, and Lasso), density estimation (MLE), dimensionality reduction (PCA and random projection), and clustering (k-means and hierarchical).

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Define and differentiate between supervised and unsupervised learning, identify instances of overfitting and underfitting in machine learning models, and perform cross-validation techniques to assess model performance.
   2. Evaluate and select appropriate algorithms for a given problem using machine learning techniques, including linear regression, logistic regression, decision trees, and k-means clustering.
   3. Apply machine learning algorithms to real-world datasets, including data preprocessing, model training, and evaluation.
   4. Develop critical thinking skills to interpret and evaluate the results of machine learning models and understand their limitations.
   5. Use popular machine learning libraries and tools such as scikit-learn, TensorFlow, and PyTorch.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Identify the fundamental concepts and terminology of machine learning, including supervised and unsupervised learning, classification, regression, and clustering.
   2. Apply commonly used machine learning algorithms and techniques, such as linear regression, logistic regression, decision trees, support vector machines, and neural networks.
   3. Apply machine learning algorithms to real-world datasets, including data preprocessing, model training, and evaluation.
   4. Apply critical thinking skills and the ability to interpret and evaluate the results of machine learning models, including understanding issues related to overfitting, underfitting, and bias.
   5. Implement popular machine learning libraries and tools such as scikit-learn, TensorFlow, and PyTorch.
   6. Utilize practical programming skills in implementing and fine-tuning machine learning models using programming languages such as Python.
   7. Examine ethical considerations in machine learning, including issues related to bias, fairness, and privacy.
   8. Employ effective communication skills for presenting and explaining machine learning concepts and results to different audiences, including technical and non-technical stakeholders.

6. **COURSE CONTENT:**
   Lecture Content:
   Topics include
   - classification: perceptrons, support vector machines (SVMs), Gaussian discriminant analysis (including linear discriminant analysis, LDA, and quadratic discriminant analysis, QDA), logistic regression, decision trees, neural networks, convolutional neural networks, boosting, nearest neighbor search;
   - regression: least-squares linear regression, logistic regression, polynomial regression, ridge regression,

Lasso;
- density estimation: maximum likelihood estimation (MLE);
- dimensionality reduction: principal components analysis (PCA), random projection; and
- clustering: *k*-means clustering, hierarchical clustering.

Lab Content:

1. Introduction to Python and Jupyter Notebooks: Introduce students to the Python programming language and the Jupyter Notebook environment, including basic Python syntax, data types, and data structures.
2. Data Preprocessing: Teach students how to preprocess datasets, including handling missing values, feature scaling, and one-hot encoding.
3. Linear Regression: Walk students through implementing a simple linear regression model using Python and applying it to a real-world dataset, including evaluating the model's performance.
4. Logistic Regression: Teach students how to implement logistic regression models using Python and apply them to binary classification problems, including evaluating the model's performance.
5. Decision Trees: Introduce students to decision trees and teach them how to build and visualize decision trees using Python, including understanding the concept of information gain and pruning.
6. K-Means Clustering: Teach students how to perform K-Means clustering on a dataset using Python, including evaluating the model's performance.
7. Neural Networks: Introduce students to neural networks and teach them how to build and train a simple neural network using Python and Keras, including understanding the concept of backpropagation.
8. Support Vector Machines: Teach students how to implement and apply support vector machines to binary and multi-class classification problems, including evaluating the model's performance.
9. Dimensionality Reduction: Teach students how to perform principal component analysis (PCA) on a dataset to reduce its dimensionality, including visualizing the reduced dataset.
10. Model Tuning and Evaluation: Walk students through the process of fine-tuning machine learning models, including hyperparameter tuning, cross-validation, and model selection, using Python and scikit-learn.

7. REPRESENTATIVE METHODS OF INSTRUCTION:

Typical methods of instruction may include:

A. Lecture
B. Lab
C. Activity
D. Critique
E. Directed Study
F. Discussion
G. Other (Specify): Lecture will be used to introduce new topics; Teacher will model problem-solving techniques. Class will solve a problem together, each person contributing a potential "step." Students will participate in short in-class projects (in teacher-organized small groups) to ensure that students experiment with the new topics in realistic problem settings; Teacher will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding; Teacher will create and manage an internet conference for discussion of course topics; and students will work in small groups on programming assignments.

8. REPRESENTATIVE ASSIGNMENTS

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

The main writing requirement for students in this course is creating documentation for their lab and programming projects. This includes both technical documentation, which outlines the problem to be solved, the project's scope, an overview of the solution, and any limitations, as well as end-user documentation, which will be provided to the client if the code is reusable.

Reading Assignments:

The course requires weekly textbook readings.

Other Outside Assignments:

The course assignments consist mainly of weekly textbook exercises and lab/programming tasks, which aim to support learning outcomes. Students will also complete several significant programs that require 500-600 lines of code.

9. REPRESENTATIVE METHODS OF EVALUATION

Representative methods of evaluation may include:

A. Exams/Tests
B. Group Projects
C. Homework
D. Lab Activities
E. Projects
F. Quizzes
G. Written examination
H. Bi-weekly quizzes (short answer-from textbook material) to provide feedback to students and teacher; assessment of student contributions during class discussion and project time; individual programming assignments; Midterm and Final exams (short answer, general problem solving (similar to in-class work), short program segments (similar to programming assignments); Assessment of group participation on course projects, including peer assessment of participation and contribution to the group effort.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Greth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An introduction to Statistical Learning*, 2 ed. Springer Science, 2021
B. Miroslav Kubat. *An Introduction to Machine Learning*, 3 ed. Springer Nature Switzerland, 2021
C. Ethem Alpaydin. *Introduction to Machine Learning*, 4 ed. The MIT Press, 2020

Origination Date: April 2023
Curriculum Committee Approval Date: September 2023
Effective Term: Fall 2024
Course Originator: Kamran Eftekhari